Top = 14
Width = 110
Height = 25
Caption = 'Create PopUp Menu'
TabOrder = 1
OnClick = Button2Click
end
<> Object Button3: TButton
Left = 200
Top = 52
Width = 110
Height = 25
Caption = 'Add PopUp Items'
Enabled = False
TabOrder = 2
OnClick = Button3Click
end
<> Object Button4: TButton
Left = 32
Top = 54
Width = 110
Height = 25
Caption = 'Add Main Items'
Enabled = False
TabOrder = 3
OnClick = Button4Click
end
<> Object Button5: TButton
Left = 32
Top = 92
Width = 110
Height = 25
Caption = 'Add Sub-Menu'
Enabled = False
TabOrder = 4
OnClick = Button5Click
end
<> Object Button6: TButton
Left = 32
Top = 130
Width = 110
Height = 25
Caption = 'Add Vertical Break'
Enabled = False
TabOrder = 5
OnClick = Button6Click
end

# القسم الأول: برمجة القوائم في دلفي

<> Object\$12
الهلف النصي لبرنامج
(Menu2)
إعداد
.1.11111 11-
علاء الدين اللباد
واته. ۷۳٬۹۷۱
<> Object Form1: TForm1
Left = 144
Top = 110
Width = 467
Height = 341
Calar - 'Menu Demo'
Color = clBtnFace Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
PixelsPerInch = 96
TextHeight = 13
<> Object Panel1: TPanel
Left = 56
Top = 56
Width = 345
Height = 209
TabOrder = 0
<> Object Button1: TButton
Left = 32 Top = 16
Width = 110
Height = 25
Caption = 'Create Main Menu'
TabOrder = 0
OnClick = Button1Click
end
<> Object Button2: TButton
Left = 200

uses
Forms,
UMnuDmo in 'UMnuDmo.pas'
{Form1};
{\$R *.RES}
(411 VIES)
begin
Application.Initialize;
Application.CreateForm(TForm1,
Form1);
Application.Run;
end.
Cita.
\\\\\\\\
unit UMnuDmo;
unit Ovinubino,
interface
Interrace
NCOC
Windows Massages SystItile
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs,
StdCtrls, ExtCtrls, Menus;
type
type TForm1 = class(TForm)
Panel1: TPanel;
,
Button1: TButton;
Button2: TButton;
Button3: TButton;
Button4: TButton;
Button5: TButton;
Button6: TButton;
Button7: TButton;
Button8: TButton;
Button9: TButton;
<> procedure
Button1Click(Sender: TObject);
<> procedure
Button2Click(Sender: TObject);
<> procedure
Button7Click(Sender: TObject);
<> procedure
Button3Click(Sender: TObject);
<> procedure

<> Object Button7: TButton
Left = 200
Top = 90
Width = 110
Height = 25
Caption = 'Show PopUp'
Enabled = False
TabOrder = 6
OnClick = Button7Click
end
<> Object Button8: TButton
Left = 32
Top = 168
Width = 110
Height = 25
Caption = 'Add Horizontal
Break'
Enabled = False
TabOrder = 7
OnClick = Button8Click
end
<> Object Button9: TButton
Left = 200
Top = 128
Width = 110
Height = 25
Caption = 'Assign OnClick
Event'
Enabled = False
TabOrder = 8
OnClick = Button9Click
end
end
End

res10	
، التنفيذ	الملة
)	
<b>भा</b> - धट	
هاتف ۳۷۱	
<b>S</b>	التفالة علاء الد التنفيذي علاء الد

(37.7.1.3.5	77 77 17 10
ShowMessage('MainMenu created	MyPopupHandler(Sender:
but no items' + #13+	TObject);
'are added so it does not	<> procedure
show.');	<b>Button4Click(Sender: TObject);</b>
<b>Button1.Enabled := False;</b>	<> procedure
end;	<b>Button5Click(Sender: TObject)</b> ;
	<> procedure
procedure	Button6Click(Sender: TObject);
TForm1.Button2Click(Sender:	<> procedure
TObject);	<b>Button8Click(Sender: TObject)</b> ;
begin	<> procedure
MyPopUpMenu :=	<b>Button9Click(Sender: TObject);</b>
TPopUpMenu.Create(self);	private
Button3.Enabled := true;	{ Private declarations }
ShowMessage('PopUpMenu	public
created.');	{ Public declarations }
Button2.Enabled := False;	end;
end;	
Ciru,	var
procedure	Form1: TForm1;
TForm1.Button7Click(Sender:	MyMainMenu: TMainMenu;
•	
TObject);	MyPopUpMenu: TPopUpMenu;
begin C. 11.6	MySubItem1, MySubItem2:
MyPopUpMenu.Popup(Form1.Left	TMenuItem;
+ 60, Form1.Top + 140);	MySubItems: array[03] of
Button9.Enabled := true;	TMenuItem;
end;	MyPopUpItems: array[03] of
	TMenuItem;
procedure	implementation
TForm1.Button3Click(Sender:	
TObject);	{\$R *.DFM}
var	
i: Integer;	procedure
begin	TForm1.MyPopupHandler(Sender:
for i := 0 to 3 do begin	TObject);
MyPopUpItems[i] :=	begin
TMenuItem.Create(Self);	with Sender as TMenuItem do
MyPopUpItems[i].Caption :=	begin
'New item ' + IntToStr(i);	ShowMessage(Caption);
	end;
MyPopUpMenu.Items.Add(MyPopU	end;
pItems[i]);	ciiu,
end;	procedure
Button7.Enabled := true;	procedure TForm 1 Putton 1 Click (Sandary
,	TForm1.Button1Click(Sender:
Button3.Enabled := False;	TObject);
end;	begin
procedure	MyMainMenu:=
TForm1.Button4Click(Sender:	TMainMenu.Create(Self);
TObject);	Button4.Enabled := true;

TForm1.Button8Click(Sender:
TObject);
begin
MySubItems[2].Caption :='-';
end;
procedure
TForm1.Button9Click(Sender:
TObject);
var
i: Integer;
begin
for i := 0 to 3 do begin
MyPopUpItems[i].OnClick :=
MyPopupHandler;
end;
end;
end.

18	
الهلف التنفيذي لبرناهج	
(Menu)	
عامدا	
علاء الدين اللباد	
واتف ۷۳۷۱۹۹۱	
unit menudemo;	
interface	
uses	
Windows, Messages, SysUtils,	
Classes, Graphics, Controls, Forms	,
Dialogs,	
Menus, ExtCtrls, ComCtrls;	
type	
TMenuForm = class(TForm)	
DisplayPanel: TPanel;	
DemoShape: TShape;	
MainMenu: TMainMenu;	
Application1: TMenuItem;	
Exit1: TMenuItem;	
Color1: TMenuItem;	

var
MyItem: array[02] of
TMenuItem;
i: Integer;
begin
for i := 0 to 2 do begin
MyItem[i] :=
TMenuItem.Create(Self);
MyItem[i].Caption := 'New item '
+ IntToStr(i);
· Int I obti (1);
MyMainMenu.Items.Add(MyItem[i]
);
end;
Button4.Enabled := False;
Button5.Enabled := true;
end;
procedure
TForm1.Button5Click(Sender:
TObject);
var
i: Integer;
begin
for i := 0 to 3 do begin
MySubItems[i] :=
TMenuItem.Create(Self);
MySubItems[i].Caption := 'New
item ' + IntToStr(i);
MySubItems[i].OnClick :=
MyPopUpHandler;
iviyi ope piranulei,
MyMainMenu.Items[0].Add(MySub
Items[i]);
end;
,
Button6.Enabled := true;
Button8.Enabled := true;
Button5.Enabled := False;
end;
procedure
TForm1.Button6Click(Sender:
TObject);
begin
MySubItems[3].Break :=
mbBarBreak;
end;
, , , , , , , , , , , , , , , , , , ,
procedure
Procedure

<> procedure	Outline1: TMenuItem;
About1Click(Sender: TObject);	Randomize1: TMenuItem;
<> procedure	Select1: TMenuItem;
Random1Click(Sender: TObject);	Inside1: TMenuItem;
<> procedure	Randomize2: TMenuItem;
RandomizeColors1Click(Sender:	Select2: TMenuItem;
TObject);	Shape1: TMenuItem;
<> procedure	Circle1: TMenuItem;
InvertColors1Click(Sender:	Ellipse1: TMenuItem;
TObject);	Rectangle1: TMenuItem;
private	Square1: TMenuItem;
{ Private declarations }	Misc1: TMenuItem;
Roundable: Boolean;	ThickOutline1: TMenuItem;
Ttounumie: Booleun,	,
function RandomColor: TColor;	RoundedShape1: TMenuItem;
Ź	Help1: TMenuItem;
<pre>&lt;&gt; procedure AlterShape(shape: TShapeType;</pre>	About1: TMenuItem;
	SolidColorDialog: TColorDialog;
roundable: Boolean);	AnyColorDialog: TColorDialog;
public	PopupMenu: TPopupMenu;
{ Public declarations }	Random1: TMenuItem;
end;	RandomizeColors1: TMenuItem;
	InvertColors1: TMenuItem;
const	<> procedure
SHAPE_OFFSET = 4;	DisplayPanelResize(Sender:
	TObject);
var	<> procedure
MenuForm: TMenuForm;	Exit1Click(Sender: TObject);
,	<> procedure
implementation	Randomize1Click(Sender: TObject);
	<> procedure
uses about;	Select1Click(Sender: TObject);
uses about,	
{\$R *.DFM}	<pre>&gt; procedure Pandamiza2Click(Sandam TObicat);</pre>
(\$R .DFWI)	Randomize2Click(Sender: TObject);
	<> procedure
,	Select2Click(Sender: TObject);
procedure	<> procedure
TMenuForm.DisplayPanelResize(Se	Circle1Click(Sender: TObject);
nder: TObject);	<> procedure
begin	Ellipse1Click(Sender: TObject);
DemoShape.Top :=	<> procedure
SHAPE_OFFSET;	Rectangle1Click(Sender: TObject);
DemoShape.Left :=	<> procedure
DemoShape.Top;	Square1Click(Sender: TObject);
DemoShape.Height :=	<> procedure
DisplayPanel.Height - 2 *	ThickOutline1Click(Sender:
SHAPE OFFSET;	TObject);
DemoShape.Width :=	<> procedure
DisplayPanel.Width - 2 *	RoundedShape1Click(Sender:
SHAPE OFFSET;	TObject);
ommin_orrowi,	1 Object),

begin	end;
DemoShape.Brush.Color :=	cnu,
RandomColor;	nuocoduno
,	procedure
end;	TMenuForm.Exit1Click(Sender:
	TObject);
procedure	begin
TMenuForm.Select2Click(Sender:	Close;
TObject);	end;
begin	
AnyColorDialog.Color :=	function
DemoShape.Brush.Color;	TMenuForm.RandomColor;
try	var
if AnyColorDialog.Execute then	red, green, blue: Byte;
DemoShape.Brush.Color :=	begin
AnyColorDialog.Color;	red := Random(255);
except	green := Random(255);
ShowMessage('Color selection	blue := Random(255);
dialog failed to load.');	bide: Random(255),
end;	Result := red or (green shl 8) or
end;	(blue shl 16);
cnu,	
nwaaaduwa TagglaChaak(Sandaw	end;
procedure ToggleCheck(Sender:	
TObject);	procedure
var	TMenuForm.Randomize1Click(Sen
Item: TMenuItem;	der: TObject);
begin	begin
Item := Sender as TMenuItem;	DemoShape.Pen.Color :=
Item.Checked := not Item.Checked;	RandomColor;
end;	end;
procedure SetCheck(Sender:	procedure
TObject);	TMenuForm.Select1Click(Sender:
var	TObject);
Item: TMenuItem;	begin
begin	try
Item := Sender as TMenuItem;	SolidColorDialog.Color :=
Item.Checked := True;	DemoShape.Pen.Color;
end;	if SolidColorDialog.Execute then
'7	DemoShape.Pen.Color :=
procedure	SolidColorDialog.Color;
TMenuForm.AlterShape(shape:	except
TShapeType; roundable: Boolean);	ShowMessage('Color selection
begin	dialog failed to load');
Self.Roundable := roundable;	end;
· ·	
DemoShape.Shape := shape;	end;
end;	
	procedure
procedure	TMenuForm.Randomize2Click(Sen
TMenuForm.Circle1Click(Sender:	der: TObject);

ToggleCheck(Sender);	TObject);
case DemoShape.Shape of	begin
stRectangle: DemoShape.Shape	SetCheck(Sender);
:= stRoundRect;	AlterShape(stCircle, False);
stRoundRect: DemoShape.Shape	end;
:= stRectangle;	
stSquare: DemoShape.Shape :=	procedure
stRoundSquare;	TMenuForm.Ellipse1Click(Sender:
stRoundSquare:	TObject);
DemoShape.Shape := stSquare;	begin
end;	SetCheck(Sender);
end;	AlterShape(stEllipse, False);
,	end;
procedure	onu,
TMenuForm.About1Click(Sender:	procedure
TObject);	TMenuForm.Rectangle1Click(Sende
var	r: TObject);
AboutBox: TAboutBox;	begin
begin	SetCheck(Sender);
AboutBox :=	` '
TAboutBox.Create(Self);	if RoundedShape1.Checked then
	AlterShape(stRoundRect, True)
try AboutBox.ShowModal;	else
	AlterShape(stRectangle, True);
finally	end;
AboutBox.Free;	
end;	procedure
end;	TMenuForm.Square1Click(Sender:
	TObject);
procedure	begin
TMenuForm.Random1Click(Sender	SetCheck(Sender);
: TObject);	if RoundedShape1.Checked then
var	AlterShape(stRoundSquare, True)
newshape: TShapeType;	else
begin	AlterShape(stSquare, True);
// Ensure we actually change the	end;
shape.	
repeat newshape :=	procedure
TShapeType(Random(6)) until	TMenuForm.ThickOutline1Click(Se
newshape <> DemoShape.Shape;	nder: TObject);
	begin
// Adjust the "Shape" and Rounded	ToggleCheck(Sender);
Shape" menu selections.	DemoShape.Pen.Width := 11 -
case newshape of	DemoShape.Pen.Width;
stEllipse: Ellipse1Click(Ellipse1);	end;
stCircle: Circle1Click(Circle1);	
stRectangle, stRoundRect:	procedure
begin	TMenuForm.RoundedShape1Click(
RoundedShape1.Checked :=	Sender: TObject);
newshape = stRoundRect;	begin
1	~ *B

Label2: TLabel;
Button1: TButton;
<> procedure
Button1Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
AboutBox: TAboutBox;
implementation
{\$R *.DFM}
procedure
TAboutBox.Button1Click(Sender:
TObject);
begin
Close;
end;
end.
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

O	bject\$
۾	الهلف النصي لبرناه
(]	Menu)
د	اعدا
د	علاء الدين اللبا
•	واتف ۹۶۶۵۷۵۳۷۱
ob	ject MenuForm: TMenuForm
L	eft = 183
T	op = 115
V	Vidth = 696
Н	leight = 480
	'aption = 'MenuForm'
C	color = clBtnFace
-	ont.Charset =

Rectangle1Click(Rectangle1);
end;
stSquare, stRoundSquare:
begin
RoundedShape1.Checked :=
newshape = stRoundSquare;
Square1Click(Square1);
end;
end;
end;
procedure
TMenuForm.RandomizeColors1Clic
k(Sender: TObject);
begin
DemoShape.Brush.Color :=
RandomColor;
DemoShape.Pen.Color :=
RandomColor;
end;
procedure
TMenuForm.InvertColors1Click(Se
nder: TObject);
var
i : Integer;
begin
i :=
Integer(DemoShape.Brush.Color)
xor \$FFFFFF;
DemoShape.Brush.Color :=
TColor(i);
end;
end.
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
unit about;
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
StdCtrls;
type
TAboutBox = class(TForm)
Label1: TLabel;

Caption = '&Outline'
Hint = 'Adjust shape outline
color'
object Randomize1:
TMenuItem
Caption = '&Randomize'
Hint = 'Set shape outline to a
random color'
OnClick = Randomize1Click
end
object Select1: TMenuItem
Caption = '&Select'
Hint = 'Select shape outline
color'
OnClick = Select1Click
end
end
object Inside1: TMenuItem
Caption = '&Inside'
Hint = 'Adjust shape inside
color'
object Randomize2:
TMenuItem
Caption = '&Randomize'
Hint = 'Set shape inside to a
random color'
OnClick = Randomize2Click
end
object Select2: TMenuItem
Caption = '&Select'
Hint = 'Select a color for the
shape filling'
OnClick = Select2Click
end
end
end
object Shape1: TMenuItem
Caption = '&Shape'
GroupIndex = 1
Hint = 'Mutually-exclusive toggle
·
group'
object Circle1: TMenuItem
Caption = '&Circle'
Checked = True
GroupIndex = 1
Hint = 'Make the shape a circle'
RadioItem = True
OnClick = Circle1Click

DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
Menu = MainMenu
OldCreateOrder = True
Position = poDefaultPosOnly
ShowHint = True
Visible = True
PixelsPerInch = 96
TextHeight = 13
object DisplayPanel: TPanel
Left = 0
Top = 0
Width = 688
Height = 426
Align = alClient
BevelInner = bvLowered
BevelOuter = bvLowered
PopupMenu = PopupMenu
TabOrder = 0
OnResize = DisplayPanelResize
object DemoShape: TShape
Left = 120
Top = 8
Width = 193
Height = 217
Shape = stCircle
end
_
end TM: M
object MainMenu: TMainMenu
Left = 56
Top = 16
object Application1: TMenuItem
Caption = '&Application'
Hint = 'Simple command menu
item'
object Exit1: TMenuItem
Caption = 'E&xit'
Hint = 'Close the application'
OnClick = Exit1Click
end
end
object Color1: TMenuItem
Caption = '&Color'
Hint = 'Submenus'
object Outline1: TMenuItem

end object Help1: TMenuItem Caption = '&Help' GroupIndex = 1 Hint = 'Simple command menu item' object About1: TMenuItem Caption = '&About' Hint = 'Summons an About box' OnClick = About1Click end end object SolidColorDialog: TColorDialog Options = [cdSolidColor] Left = 56 Top = 72 end object AnyColorDialog: TColorDialog Options = [cdAnyColor] Left = 56 Top = 208 end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object Randomize Colors1: TMenuItem Caption = 'Randomize &Colors' OnClick = Randomize Colors' OnClick = InvertColors1 Click end end End	RoundedShape1Click
object Help1: TMenuItem Caption = '&Help' GroupIndex = 1 Hint = 'Simple command menu item' object About1: TMenuItem Caption = '&About' Hint = 'Summons an About box' OnClick = About1Click end end object SolidColorDialog: TColorDialog Options = [cdSolidColor] Left = 56 Top = 72 end object AnyColorDialog: TColorDialog Options = [cdAnyColor] Left = 56 Top = 208 end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1 TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	end
Caption = '&Help' GroupIndex = 1 Hint = 'Simple command menu item' object About1: TMenuItem Caption = '&About' Hint = 'Summons an About box' OnClick = About1Click end end object SolidColorDialog: TColorDialog Options = [cdSolidColor] Left = 56 Top = 72 end object AnyColorDialog: TColorDialog Options = [cdAnyColor] Left = 56 Top = 208 end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = InvertColors1Click end end	end
GroupIndex = 1  Hint = 'Simple command menu item'  object About1: TMenuItem  Caption = '&About'  Hint = 'Summons an About box'  OnClick = About1Click  end  end  object SolidColorDialog: TColorDialog  Options = [cdSolidColor]  Left = 56  Top = 72  end  object AnyColorDialog: TColorDialog  Options = [cdAnyColor]  Left = 56  Top = 208  end  object PopupMenu: TPopupMenu  Left = 64  Top = 144  object Random1: TMenuItem  Caption = 'Randomize &Shape'  OnClick = Random1Click  end  object RandomizeColors1: TMenuItem  Caption = 'Randomize &Colors'  OnClick = RandomizeColors1: TMenuItem  Caption = 'Randomize &Colors'  OnClick = RandomizeColors1: TMenuItem  Caption = 'Randomize &Colors'  OnClick = RandomizeColors1: TMenuItem  Caption = '&Invert Inside Color'  OnClick = InvertColors1Click  end  end	object Help1: TMenuItem
Hint = 'Simple command menu item'  object About1: TMenuItem Caption = '&About'  Hint = 'Summons an About box' OnClick = About1Click end end object SolidColorDialog: TColorDialog Options = [cdSolidColor]  Left = 56 Top = 72 end object AnyColorDialog: TColorDialog Options = [cdAnyColor]  Left = 56 Top = 208 end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1: TMenuItem Caption = 'RandomizeColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	Caption = '&Help'
item' Object About1: TMenuItem Caption = '&About' Hint = 'Summons an About box' OnClick = About1Click end end object SolidColorDialog: TColorDialog Options = [cdSolidColor] Left = 56 Top = 72 end object AnyColorDialog: TColorDialog Options = [cdAnyColor] Left = 56 Top = 208 end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1: TMenuItem Caption = '&InvertColors1: TMenuItem Caption = '&InvertInside Color' OnClick = InvertColors1Click end end	GroupIndex = 1
object About1: TMenuItem Caption = '&About' Hint = 'Summons an About box' OnClick = About1Click end end object SolidColorDialog: TColorDialog Options = [cdSolidColor] Left = 56 Top = 72 end object AnyColorDialog: TColorDialog Options = [cdAnyColor] Left = 56 Top = 208 end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1: TMenuItem Caption = '&InvertColors1: TMenuItem Caption = '&InvertInside Color' OnClick = InvertColors1Click end end	Hint = 'Simple command menu
Caption = '&About'  Hint = 'Summons an About box' OnClick = About1Click end end end object SolidColorDialog: TColorDialog Options = [cdSolidColor] Left = 56 Top = 72 end object AnyColorDialog: TColorDialog Options = [cdAnyColor] Left = 56 Top = 208 end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	item'
Hint = 'Summons an About box' OnClick = About1Click end end end object SolidColorDialog: TColorDialog Options = [cdSolidColor] Left = 56 Top = 72 end object AnyColorDialog: TColorDialog Options = [cdAnyColor] Left = 56 Top = 208 end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	object About1: TMenuItem
OnClick = About1Click end end object SolidColorDialog: TColorDialog Options = [cdSolidColor] Left = 56 Top = 72 end object AnyColorDialog: TColorDialog Options = [cdAnyColor] Left = 56 Top = 208 end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	Caption = '&About'
end end object SolidColorDialog: TColorDialog Options = [cdSolidColor] Left = 56 Top = 72 end object AnyColorDialog: TColorDialog Options = [cdAnyColor] Left = 56 Top = 208 end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	Hint = 'Summons an About box'
end object SolidColorDialog: TColorDialog Options = [cdSolidColor] Left = 56 Top = 72 end object AnyColorDialog: TColorDialog Options = [cdAnyColor] Left = 56 Top = 208 end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	OnClick = About1Click
end object SolidColorDialog: TColorDialog Options = [cdSolidColor] Left = 56 Top = 72 end object AnyColorDialog: TColorDialog Options = [cdAnyColor] Left = 56 Top = 208 end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	end
object SolidColorDialog: TColorDialog Options = [cdSolidColor] Left = 56 Top = 72 end object AnyColorDialog: TColorDialog Options = [cdAnyColor] Left = 56 Top = 208 end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	end
Options = [cdSolidColor]  Left = 56  Top = 72 end object AnyColorDialog: TColorDialog Options = [cdAnyColor]  Left = 56  Top = 208 end object PopupMenu: TPopupMenu  Left = 64  Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	end
Options = [cdSolidColor]  Left = 56  Top = 72 end object AnyColorDialog: TColorDialog Options = [cdAnyColor]  Left = 56  Top = 208 end object PopupMenu: TPopupMenu  Left = 64  Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	object SolidColorDialog:
Left = 56 Top = 72 end object AnyColorDialog: TColorDialog Options = [cdAnyColor] Left = 56 Top = 208 end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	
Left = 56 Top = 72 end object AnyColorDialog: TColorDialog Options = [cdAnyColor] Left = 56 Top = 208 end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	Options = [cdSolidColor]
end object AnyColorDialog: TColorDialog Options = [cdAnyColor] Left = 56 Top = 208 end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	
object AnyColorDialog: TColorDialog Options = [cdAnyColor] Left = 56 Top = 208 end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	Top = 72
Options = [cdAnyColor]  Left = 56  Top = 208 end object PopupMenu: TPopupMenu  Left = 64  Top = 144 object Random1: TMenuItem  Caption = 'Randomize & Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem  Caption = 'Randomize & Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '& Invert Inside Color' OnClick = InvertColors1Click end end	end
Options = [cdAnyColor]  Left = 56  Top = 208 end object PopupMenu: TPopupMenu  Left = 64  Top = 144 object Random1: TMenuItem  Caption = 'Randomize & Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem  Caption = 'Randomize & Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '& Invert Inside Color' OnClick = InvertColors1Click end end	object AnyColorDialog:
Left = 56 Top = 208 end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	TColorDialog
Top = 208 end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	Options = [cdAnyColor]
end object PopupMenu: TPopupMenu Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	Left = 56
object PopupMenu: TPopupMenu  Left = 64  Top = 144  object Random1: TMenuItem  Caption = 'Randomize &Shape'  OnClick = Random1Click  end  object RandomizeColors1:  TMenuItem  Caption = 'Randomize &Colors'  OnClick =  RandomizeColors1Click  end  object InvertColors1: TMenuItem  Caption = '&Invert Inside Color'  OnClick = InvertColors1Click  end  end	Top = 208
Left = 64 Top = 144 object Random1: TMenuItem Caption = 'Randomize & Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize & Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '& Invert Inside Color' OnClick = InvertColors1Click end end	end
Top = 144 object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	object PopupMenu: TPopupMenu
object Random1: TMenuItem Caption = 'Randomize &Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	Left = 64
Caption = 'Randomize & Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize & Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '& Invert Inside Color' OnClick = InvertColors1Click end end	Top = 144
Caption = 'Randomize & Shape' OnClick = Random1Click end object RandomizeColors1: TMenuItem Caption = 'Randomize & Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '& Invert Inside Color' OnClick = InvertColors1Click end end	object Random1: TMenuItem
end object RandomizeColors1: TMenuItem Caption = 'Randomize & Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '& Invert Inside Color' OnClick = InvertColors1Click end end	
object RandomizeColors1: TMenuItem Caption = 'Randomize &Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	OnClick = Random1Click
TMenuItem Caption = 'Randomize & Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '& Invert Inside Color' OnClick = InvertColors1Click end end	end
Caption = 'Randomize & Colors' OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '& Invert Inside Color' OnClick = InvertColors1Click end end	object RandomizeColors1:
OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	TMenuItem
OnClick = RandomizeColors1Click end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	Caption = 'Randomize & Colors'
end object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	
object InvertColors1: TMenuItem Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	RandomizeColors1Click
Caption = '&Invert Inside Color' OnClick = InvertColors1Click end end	end
OnClick = InvertColors1Click end end	object InvertColors1: TMenuItem
OnClick = InvertColors1Click end end	
end end	
End	end
	End
object AboutBox: TAboutBox	object AboutBox: TAboutBox

end
object Ellipse1: TMenuItem
Caption = '&Ellipse'
GroupIndex = 1
Hint = 'Make the shape an
ellipse'
RadioItem = True
OnClick = Ellipse1Click
end
object Rectangle1: TMenuItem
Caption = '&Rectangle'
GroupIndex = 1
Hint = 'Make the shape a
rectangle'
RadioItem = True
OnClick = Rectangle1Click
end
object Square1: TMenuItem
Caption = '□'
GroupIndex = 1
Hint = 'Make the shape a
square'
RadioItem = True
OnClick = Square1Click
end
end
object Misc1: TMenuItem
Caption = '&Misc'
GroupIndex = 1
Hint =
'Simple checkable menu items
with a superfluous bar break in
betw' +
'een'
object ThickOutline1:
TMenuItem
Caption = '&Thick Outline'
Hint = 'Controls whether or not
the outline is thick'
OnClick = ThickOutline1Click
end
object RoundedShape1:
TMenuItem
Break = mbBarBreak
Caption = '&Rounded Shape'
Hint = 'Makes rectangle or
square have rounded corners'
OnClick =
Onence -

end	
End	
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	

Procedure\$5
الهلف التنفيذي لبرناهج
(TabCntrl)
إعداد
علاء الدين اللباد
واتف ۷۳۲۵۷۵۴۹۰
program clsexm;
USES Forms
Forms,
clsexm1 in 'clsexm1.pas' {Form1};
{\$R *.RES}
(wit itels)
begin
Application.Initialize;
Application.CreateForm(TForm1,
Form1);
Application.Run;
end.
wit drawn1
unit clsexm1;
interface
interrace
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
Db, DBTables, ComCtrls, ExtCtrls,
DBCtrls, StdCtrls, Mask;
type
TForm1 = class(TForm)
TabControl1: TTabControl;
Table1: TTable;
DataSource1: TDataSource;
Table1EmpNo: TIntegerField;
Table1LastName: TStringField;

Left = 189
Top = 110
BorderIcons = []
BorderStyle = bsDialog
Caption = 'About Menu'
ClientHeight = 134
ClientWidth = 239
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
PixelsPerInch = 96
TextHeight = 13
object Label1: TLabel
Left = 60
Top = 24
Width = 123
Height = 16
Alignment = taCenter
U
Cantian = 'Manu Demonstration'
Caption = 'Menu Demonstration'  Font Charset = ANSL CHARSET
Font.Charset = ANSI_CHARSET
Font.Charset = ANSI_CHARSET Font.Color = clBlack
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif'
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = []
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end object Label2: TLabel
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end object Label2: TLabel Left = 32
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end object Label2: TLabel Left = 32 Top = 56
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end object Label2: TLabel Left = 32 Top = 56 Width = 177
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end object Label2: TLabel Left = 32 Top = 56 Width = 177 Height = 13
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end object Label2: TLabel Left = 32 Top = 56 Width = 177 Height = 13 Caption = 'Copyright (c) 1999
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end object Label2: TLabel Left = 32 Top = 56 Width = 177 Height = 13 Caption = 'Copyright (c) 1999 Inprise Corporation'
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end object Label2: TLabel Left = 32 Top = 56 Width = 177 Height = 13 Caption = 'Copyright (c) 1999 Inprise Corporation' end
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end object Label2: TLabel Left = 32 Top = 56 Width = 177 Height = 13 Caption = 'Copyright (c) 1999 Inprise Corporation' end object Button1: TButton
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end object Label2: TLabel Left = 32 Top = 56 Width = 177 Height = 13 Caption = 'Copyright (c) 1999 Inprise Corporation' end object Button1: TButton Left = 80
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end object Label2: TLabel Left = 32 Top = 56 Width = 177 Height = 13 Caption = 'Copyright (c) 1999 Inprise Corporation' end object Button1: TButton Left = 80 Top = 88
Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end object Label2: TLabel Left = 32 Top = 56 Width = 177 Height = 13 Caption = 'Copyright (c) 1999 Inprise Corporation' end object Button1: TButton Left = 80 Top = 88 Width = 75
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end object Label2: TLabel Left = 32 Top = 56 Width = 177 Height = 13 Caption = 'Copyright (c) 1999 Inprise Corporation' end object Button1: TButton Left = 80 Top = 88 Width = 75 Height = 25
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end object Label2: TLabel Left = 32 Top = 56 Width = 177 Height = 13 Caption = 'Copyright (c) 1999 Inprise Corporation' end object Button1: TButton Left = 80 Top = 88 Width = 75 Height = 25 Caption = 'OK'
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end object Label2: TLabel Left = 32 Top = 56 Width = 177 Height = 13 Caption = 'Copyright (c) 1999 Inprise Corporation' end object Button1: TButton Left = 80 Top = 88 Width = 75 Height = 25

begin  TabControl1.Tabs.Add(FieldByNam e('LASTNAME').AsString);  Next; end; // Sync page and table. First; end; end; end;
e('LASTNAME').AsString);  Next; end; // Sync page and table. First; end; end;
Next; end; // Sync page and table. First; end; end;
Next; end; // Sync page and table. First; end; end;
// Sync page and table. First; end; end;
// Sync page and table. First; end; end;
First; end; end;
end;
<> procedure
TForm1.TabControl1Change(Sende
r: TObject);
begin
with TabControl1 do
begin
if TabIndex <> -1 then
if not
Table1.Locate('LASTNAME',
Tabs[TabIndex], []) then
ShowMessage(Tabs[TabIndex -
1] + ' not found.');
end;
end;
<> procedure
TForm1.Table1CalcFields(DataSet:
TDataSet);
begin
DataSet['FullName'] :=
DataSet['LASTNAME'] + ', ' +
DataSet['FIRSTNAME'];
end;
end.

Table1FirstName: TStringField;
Tablett it suvaine. Tsu ingrieiu,
Table1PhoneExt: TStringField;
Table1HireDate:
TDateTimeField;
Table1Salary: TFloatField;
Label1: TLabel;
DBEdit1: TDBEdit;
Label2: TLabel;
DBEdit2: TDBEdit;
Label3: TLabel;
DBEdit3: TDBEdit;
Label4: TLabel;
DBEdit4: TDBEdit;
Label5: TLabel;
DBEdit5: TDBEdit;
Label6: TLabel;
DBEdit6: TDBEdit;
DBText1: TDBText;
,
Table1FullName: TStringField;
r - s - s - s - s
FormCreate(Sender: TObject);
<> procedure
TabControl1Change(Sender:
TObject); <> procedure
Table1CalcFields(DataSet:
TDataSet);
end;
cnu,
YOM
Var
Form1: TForm1;
:
implementation
(OD 4 DELO
{\$R *.DFM}
<> procedure
TForm1.FormCreate(Sender:
TObject);
begin
TabControl1.Align := alClient;
TabControl1.TabHeight := 30;
with Table1 do
begin
begin Close;
Close;

Height = 13 Caption = 'LastName' FocusControl = DBEdit2 end object Label3: TLabel Left = 16
FocusControl = DBEdit2 end object Label3: TLabel Left = 16
end object Label3: TLabel Left = 16
object Label3: TLabel Left = 16
Left = 16
Top = 152
Width = 47
Height = 13
Caption = 'FirstName'
FocusControl = DBEdit3
end
object Label4: TLabel
Left = 16
Top = 192 Width = 46
Height = 13
Caption = 'PhoneExt'
FocusControl = DBEdit4
end
object Label5: TLabel
Left = 16
Top = 232
Width = 42
Height = 13
Caption = 'HireDate'
FocusControl = DBEdit5
end
object Label6: TLabel
Left = 16
Top = 272
Width = 29
Height = 13
Caption = 'Salary'
FocusControl = DBEdit6
end
object DBText1: TDBText
Left = 56
Top = 48
Width = 6
Height = 20
AutoSize = True
DataField = 'FullName'
DataSource = DataSource1
Font.Charset =
ANSI_CHARSET
Font.Color = clBlack
Font.Height = -16

(TabCntrl)  (TabCntrl)  Alas    Alas		
object Form1: TForm1  Left = 292  Top = 153  Width = 366  Height = 354  Caption = 'Form1'  Color = clBtnFace  Font.Charset =  DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  OldCreateOrder = True  OnCreate = FormCreate  PixelsPerInch = 96  TextHeight = 13  object TabControl1: TTabControl  Left = 0  Top = 0  Width = 358  Height = 320  Align = alClient  TabOrder = 0  OnChange = TabControl1Change object Label1: TLabel  Left = 16  Top = 72  Width = 35  Height = 13  Caption = 'EmpNo'  FocusControl = DBEdit1  end object Label2: TLabel  Left = 16		الهلف النصي لبرناهج
object Form1: TForm1  Left = 292  Top = 153  Width = 366  Height = 354  Caption = 'Form1'  Color = clBtnFace  Font.Charset =  DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  OldCreateOrder = True  OnCreate = FormCreate  PixelsPerInch = 96  TextHeight = 13  object TabControl1: TTabControl  Left = 0  Top = 0  Width = 358  Height = 320  Align = alClient  TabOrder = 0  OnChange = TabControl1Change  object Label1: TLabel  Left = 16  Top = 72  Width = 35  Height = 13  Caption = 'EmpNo'  FocusControl = DBEdit1  end  object Label2: TLabel  Left = 16		(TabCntrl)
object Form1: TForm1  Left = 292  Top = 153  Width = 366  Height = 354  Caption = 'Form1'  Color = clBtnFace  Font.Charset =  DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  OldCreateOrder = True  OnCreate = FormCreate  PixelsPerInch = 96  TextHeight = 13  object TabControl1: TTabControl  Left = 0  Top = 0  Width = 358  Height = 320  Align = alClient  TabOrder = 0  OnChange = TabControl1Change  object Label1: TLabel  Left = 16  Top = 72  Width = 35  Height = 13  Caption = 'EmpNo'  FocusControl = DBEdit1  end  object Label2: TLabel  Left = 16		إعداد
object Form1: TForm1  Left = 292  Top = 153  Width = 366  Height = 354  Caption = 'Form1'  Color = clBtnFace  Font.Charset =  DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  OldCreateOrder = True  OnCreate = FormCreate  PixelsPerInch = 96  TextHeight = 13  object TabControl1: TTabControl  Left = 0  Top = 0  Width = 358  Height = 320  Align = alClient  TabOrder = 0  OnChange = TabControl1Change  object Label1: TLabel  Left = 16  Top = 72  Width = 35  Height = 13  Caption = 'EmpNo'  FocusControl = DBEdit1  end  object Label2: TLabel  Left = 16		علاء الدين اللباد
Top = 153 Width = 366 Height = 354 Caption = 'Form1' Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Reight = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 object TabControl1: TTabControl Left = 0 Top = 0 Width = 358 Height = 320 Align = alClient TabOrder = 0 OnChange = TabControl1Change object Label1: TLabel Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		واتف ۷۳۷۵۷۵۶۹۰
Top = 153 Width = 366 Height = 354 Caption = 'Form1' Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Color = clWindowText Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 object TabControl1: TTabControl Left = 0 Top = 0 Width = 358 Height = 320 Align = alClient TabOrder = 0 OnChange = TabControl1Change object Label1: TLabel Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		•
Width = 366 Height = 354 Caption = 'Form1' Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Color = clWindowText Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 object TabControl1: TTabControl Left = 0 Top = 0 Width = 358 Height = 320 Align = alClient TabOrder = 0 OnChange = TabControl1Change object Label1: TLabel Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		
Height = 354 Caption = 'Form1' Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Color = clWindowText Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 object TabControl1: TTabControl Left = 0 Top = 0 Width = 358 Height = 320 Align = alClient TabOrder = 0 OnChange = TabControl1Change object Label1: TLabel Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		•
Caption = 'Form1' Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Color = clWindowText Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 object TabControl1: TTabControl Left = 0 Top = 0 Width = 358 Height = 320 Align = alClient TabOrder = 0 OnChange = TabControl1Change object Label1: TLabel Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		
Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Color = clWindowText Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 object TabControl1: TTabControl Left = 0 Top = 0 Width = 358 Height = 320 Align = alClient TabOrder = 0 OnChange = TabControl1Change object Label1: TLabel Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		
Font.Charset = DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = [] OldCreateOrder = True OnCreate = FormCreate  PixelsPerInch = 96  TextHeight = 13 object TabControl1: TTabControl  Left = 0  Top = 0  Width = 358  Height = 320  Align = alClient  TabOrder = 0 OnChange = TabControl1Change object Label1: TLabel  Left = 16  Top = 72  Width = 35  Height = 13  Caption = 'EmpNo'  FocusControl = DBEdit1  end object Label2: TLabel  Left = 16		
DEFAULT_CHARSET Font.Color = clWindowText Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 object TabControl1: TTabControl Left = 0 Top = 0 Width = 358 Height = 320 Align = alClient TabOrder = 0 OnChange = TabControl1Change object Label1: TLabel Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		
Font.Color = clWindowText Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 object TabControl1: TTabControl Left = 0 Top = 0 Width = 358 Height = 320 Align = alClient TabOrder = 0 OnChange = TabControl1Change object Label1: TLabel Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		
Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 object TabControl1: TTabControl Left = 0 Top = 0 Width = 358 Height = 320 Align = alClient TabOrder = 0 OnChange = TabControl1Change object Label1: TLabel Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		
Font.Name = 'MS Sans Serif'  Font.Style = []  OldCreateOrder = True  OnCreate = FormCreate  PixelsPerInch = 96  TextHeight = 13  object TabControl1: TTabControl  Left = 0  Top = 0  Width = 358  Height = 320  Align = alClient  TabOrder = 0  OnChange = TabControl1Change object Label1: TLabel  Left = 16  Top = 72  Width = 35  Height = 13  Caption = 'EmpNo'  FocusControl = DBEdit1  end object Label2: TLabel  Left = 16		
Font.Style = [] OldCreateOrder = True OnCreate = FormCreate PixelsPerInch = 96  TextHeight = 13 object TabControl1: TTabControl Left = 0 Top = 0 Width = 358 Height = 320 Align = alClient TabOrder = 0 OnChange = TabControl1Change object Label1: TLabel Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		
OldCreateOrder = True OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 object TabControl1: TTabControl Left = 0 Top = 0 Width = 358 Height = 320 Align = alClient TabOrder = 0 OnChange = TabControl1Change object Label1: TLabel Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		
PixelsPerInch = 96  TextHeight = 13  object TabControl1: TTabControl  Left = 0  Top = 0  Width = 358  Height = 320  Align = alClient  TabOrder = 0  OnChange = TabControl1Change object Label1: TLabel  Left = 16  Top = 72  Width = 35  Height = 13  Caption = 'EmpNo'  FocusControl = DBEdit1  end object Label2: TLabel  Left = 16		V 11
TextHeight = 13 object TabControl1: TTabControl Left = 0 Top = 0 Width = 358 Height = 320 Align = alClient TabOrder = 0 OnChange = TabControl1Change object Label1: TLabel Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		
object TabControl1: TTabControl Left = 0 Top = 0 Width = 358 Height = 320 Align = alClient TabOrder = 0 OnChange = TabControl1Change object Label1: TLabel Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16	]	PixelsPerInch = 96
Left = 0 Top = 0 Width = 358 Height = 320 Align = alClient TabOrder = 0 OnChange = TabControl1Change object Label1: TLabel Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		
Top = 0  Width = 358  Height = 320  Align = alClient  TabOrder = 0  OnChange = TabControl1Change object Label1: TLabel  Left = 16  Top = 72  Width = 35  Height = 13  Caption = 'EmpNo'  FocusControl = DBEdit1 end object Label2: TLabel  Left = 16	•	
Width = 358 Height = 320 Align = alClient TabOrder = 0 OnChange = TabControl1Change object Label1: TLabel Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		
Height = 320 Align = alClient TabOrder = 0 OnChange = TabControl1Change object Label1: TLabel Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		
Align = alClient TabOrder = 0 OnChange = TabControl1Change object Label1: TLabel Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		
TabOrder = 0 OnChange = TabControl1Change object Label1: TLabel Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		-
OnChange = TabControl1Change object Label1: TLabel Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		C
object Label1: TLabel  Left = 16  Top = 72  Width = 35  Height = 13  Caption = 'EmpNo'  FocusControl = DBEdit1  end object Label2: TLabel  Left = 16		
Left = 16 Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		
Top = 72 Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		
Width = 35 Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		
Height = 13 Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		
Caption = 'EmpNo' FocusControl = DBEdit1 end object Label2: TLabel Left = 16		
FocusControl = DBEdit1 end object Label2: TLabel Left = 16		
object Label2: TLabel Left = 16		
Left = 16		end
Ton = 112		
TVP III		Top = 112

Width = 48

end
object DBEdit6: TDBEdit
Left = 16
Top = 288
Width = 64
Height = 21
DataField = 'Salary'
DataSource = DataSource1
TabOrder = 5
end
end
object Table1: TTable
Active = True
OnCalcFields = Table1CalcFields
DatabaseName = 'DBDEMOS'
TableName = 'EMPLOYEE.DB'
Left = 16
Top = 16
object Table1EmpNo:
TIntegerField
FieldName = 'EmpNo'
DisplayFormat = 'Emp''#'' 0000'
MaxValue = 9999
MinValue = 1
end
object Table1LastName:
TStringField
FieldName = 'LastName'
end
object Table1FirstName:
TStringField
FieldName = 'FirstName'
Size = 15
end
object Table1PhoneExt:
TStringField
FieldName = 'PhoneExt'
Size = 4
end
object Table1HireDate:
TDateTimeField
FieldName = 'HireDate'
end
object Table1Salary: TFloatField
FieldName = 'Salary'
end
object Table1FullName:
TStringField

Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
object DBEdit1: TDBEdit
Left = 16
Top = 88
Width = 64
Height = 21
DataField = 'EmpNo'
DataSource = DataSource1
TabOrder = 0
end
object DBEdit2: TDBEdit
Left = 16
Top = 128
Width = 124
Height = 21
DataField = 'LastName'
DataSource = DataSource1
TabOrder = 1
end
object DBEdit3: TDBEdit
Left = 16
Top = 168
Width = 94
Height = 21
DataField = 'FirstName'
DataSource = DataSource1
TabOrder = 2
end
object DBEdit4: TDBEdit
Left = 16
Top = 208
Width = 28
Height = 21
DataField = 'PhoneExt'
DataSource = DataSource1
TabOrder = 3
end
object DBEdit5: TDBEdit
Left = 16
Top = 248
<b>Width</b> = 64
Height = 21
DataField = 'HireDate'
DataSource = DataSource1
TabOrder = 4
I WOOT WOT

{\$R *.res}
begin
Application.Initialize;
Application.finitianze,
Application.CreateForm(TTextBrow serForm, TextBrowserForm);
Application.Run;
end.
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
unit main;
intonfo o
interface
waa
SysUtils, Types, Classes,
QGraphics, QControls, QForms,
QDialogs,
QComCtrls, QStdCtrls, QExtCtrls,
QButtons;
,
type
TTextBrowserForm =
class(TForm)
HomeBtn: TButton;
BackBtn: TButton;
ForwardBtn: TButton;
TextBrowser: TTextBrowser;
GroupBox1: TGroupBox;
RadioButton1: TRadioButton;
RadioButton2: TRadioButton;
Label1: TLabel;
LinkSourceEdit: TEdit;
ReloadBtn: TButton; GroupBox2: TGroupBox;
RadioButton3: TRadioButton;
RadioButton4: TRadioButton;
RadioGroup1: TRadioGroup;
UnderlineCheckbox: TCheckBox;
StatusBar: TStatusBar;
SpeedButton1: TSpeedButton;
<> procedure
HomeBtnClick(Sender: T<
>Object);
<> procedure
TextBrowserTextChanged(Sender:

FieldKind = fkCalculated
FieldName = 'FullName'
Size = 30
Calculated = True
end
end
object DataSource1: TDataSource
DataSet = Table1
Left = 56
Top = 24
end
End

# القسم الثاني: برامج النصوص في دلفي

<u> </u>	
Procedures 32	
الملف التنفيذي لبرنامج	
(TextBrowser)	
عامد إ	
علاء الدين اللباد	
هاتف ۷۱۳۵۷۵۶۹۰	
الملف التنفيذي للبرنامج	
TextBrowser;	
ويوجد فيه	
Procedures 32	
إعداد:	
مركز علاء الدين للمعلوماتية	
علاء الدين محمد اللباد	
.9 £ £ 0 V 0 T V 1	
program TextBrowser;	
uses	
QForms,	
main in 'main.pas'	
{TextBrowserForm};	

{ Public declarations }	T<>Obje
end;	<> pro
	BackBtnClick
var	>Object);
TextBrowserForm:	<> pro
TTextBrowserForm;	ForwardBtnC
	>Object);
implementation	<> pro
	RadioButton1
{\$R *.xfm}	>Object);
	<> pro
const	RadioButton2
DefaultDocName = 'index.html';	>Object);
FormCaption = 'TTextBrowser	<> pro
Demo Application - ';	LinkSourceEd
	>Object);
<> procedure	<> proc
TTextBrowserForm.FormCreate(Se	FormCreate(S
nder: T<>Object);	>Object);
begin	<> pro
PathList := TStringList.Create;	FormDestroy(
LinkSourceEdit.Text :=	>Object);
ExtractFilePath(Application.ExeNa	<> pro
me);	ReloadBtnClid
TextBrowser.FileName :=	>Object);
LinkSourceEdit.Text +	<> prog RadioButton3
DefaultDocName;	>Object);
UnderlineCheckbox.Checked := not	<> pro
TextBrowser.UnderlineLink;	RadioButton4
end;	>Object);
	<> pro
<> procedure	RadioGroup1
TTextBrowserForm.HomeBtnClick(	>Object);
Sender: T<>Object);	<> pro
begin	UnderlineChe
TextBrowser.Home;	T<>Obje
end;	<> pro
	TextBrowserH
<> procedure	T<>Obje
TTextBrowserForm.TextBrowserTe	const High
xtChanged(Sender: T<	WideString);
>Object); begin	<> pro
BackBtn.Enabled :=	SpeedButton1
TextBrowser.CanGoBackward;	>Object);
ForwardBtn.Enabled :=	private
TextBrowser.CanGoForward;	{ Private de
Caption := FormCaption +	PathList: T
TextBrowser.DocumentTitle;	public

T<>Object);
<> procedure
BackBtnClick(Sender: T<
>Object);
<> procedure
ForwardBtnClick(Sender: T<
>Object);
<> procedure
RadioButton1Click(Sender: T<
>Object);
<> procedure
RadioButton2Click(Sender: T<
>Object);
<pre>&lt;&gt; procedure LinkSourceEditChange(Sandaw T</pre>
LinkSourceEditChange(Sender: T<-
>Object);
<> procedure
FormCreate(Sender: T<
>Object);
<> procedure
FormDestroy(Sender: T<
>Object);
<> procedure
ReloadBtnClick(Sender: T<
>Object);
<> procedure
RadioButton3Click(Sender: T<
>Object);
<> procedure
RadioButton4Click(Sender: T<
>Object);
<> procedure
RadioGroup1Click(Sender: T<
>Object);
<> procedure
UnderlineCheckboxClick(Sender:
T<>Object);
<> procedure
TextBrowserHighlightText(Sender:
T<>Object;
const HighlightedText:
WideString);
<> procedure
SpeedButton1Click(Sender: T<
>Object);
private
{ Private declarations }
PathList: TStrings;
public
· •

TTextBrowserForm.ReloadBtnClick	end;
(Sender: T<>Object);	
begin	<> procedure
	TTextBrowserForm.BackBtnClick(S
TextBrowser.LoadFromFile(TextBr	ender: T<>Object);
owser.FileName);	begin
end;	TextBrowser.Backward;
	end;
<> procedure	
TTextBrowserForm.RadioButton3C	<> procedure
lick(Sender: T<>Object);	TTextBrowserForm.ForwardBtnCli
begin	ck(Sender: T<>Object);
TextBrowser.TextColor := clBlack;	begin
end;	TextBrowser.Forward;
	end;
<> procedure	,
TTextBrowserForm.RadioButton4C	<> procedure
lick(Sender: T<>Object);	TTextBrowserForm.RadioButton1C
begin	lick(Sender: T<>Object);
TextBrowser.TextColor := clGreen;	begin
end;	TextBrowser.LinkColor := clBlue;
	end;
<> procedure	,
TTextBrowserForm.RadioGroup1Cl	<> procedure
ick(Sender: T<>Object);	TTextBrowserForm.RadioButton2C
begin	lick(Sender: T<>Object);
case RadioGroup1.ItemIndex of	begin
0: TextBrowser.TextFormat :=	TextBrowser.LinkColor := clRed;
tfText;	end;
1: TextBrowser.TextFormat :=	
tfPlainText;	<> procedure
end;	TTextBrowserForm.LinkSourceEdit
end;	Change(Sender: T<>Object);
	begin
<> procedure	
TTextBrowserForm.UnderlineChec	PathList.Add(LinkSourceEdit.Text);
kboxClick(Sender: T<	TextBrowser.Factory.FilePath :=
>Object);	PathList;
begin	end;
TextBrowser.UnderlineLink := not	
TextBrowser.UnderlineLink;	
end;	<> procedure
	TTextBrowserForm.FormDestroy(S
<> procedure	ender: T<>Object);
TTextBrowserForm.TextBrowserHi	begin
ghlightText(Sender: T<	PathList.Free;
>Object;	end;
const HighlightedText:	
WideString);	<> procedure

Left = 600
Top = 16
Width = 23
Height = 22
Anchors = [akTop, akRight]
Glyph.Data = {
}
NumGlyphs = 2
OnClick = SpeedButton1Click
end
<> object HomeBtn: TButton
Left = 8
Top = 64
Width = 75
Height = 25
Caption = 'Home'
TabOrder = 5
OnClick = HomeBtnClick
end
<> object BackBtn: TButton
Left = 88
Top = 64
Width = 75
Height = 25
Caption = 'Back'
TabOrder = 0
OnClick = BackBtnClick
end
<> object ForwardBtn:
TButton
Left = 168
Top = 64
Width = 75
Height = 25
Caption = 'Forward'
TabOrder = 1
OnClick = ForwardBtnClick
end
<> object TextBrowser:
TTextBrowser
Left = 8
Top = 103
Width = 624
Height = 334
Anchors = [akLeft, akTop,
akRight, akBottom]
TabOrder = 2
TextFormat = tfText

begin
StatusBar.SimpleText := 'Link To:
' + HighlightedText;
end;
<> procedure
TTextBrowserForm.SpeedButton1C
lick(Sender: T<>Object);
var
NewDir: WideString;
begin
if SelectDirectory('Select a
Directory', ", NewDir) then
LinkSourceEdit.Text := NewDir;
end;
end.

Object 18
الملف النصى للبرنامج
TextBrowse
وفيه ۱۸ اجراءا
اعداد
علاء الدين اللباد
<u>094575371</u>
2 11 4 m (D D
<> object TextBrowserForm:
TTextBrowserForm
Left = 183
Top = 131
Width = 640
Height = 469
VertScrollBar.Range = 119
HorzScrollBar.Range = 633
ActiveControl = BackBtn
Caption = 'TTextBrowser
Component Demo Application'
Color = clBackground
OnCreate = FormCreate
OnDestroy = FormDestroy
PixelsPerInch = 75
TextHeight = 13
TextWidth = 6
<> object SpeedButton1:
TSpeedButton

J
end
<> object Label1: TLabel
Left = 280
Top = 1
Width = 82
Height = 13
Caption = 'Link Source Path'
end
<> object LinkSourceEdit: TEdit
Left = 280
Top = 16
Width = 313
Height = 21
Anchors = [akLeft, akTop,
akRight]
TabOrder = 6
OnChange =
LinkSourceEditChange
end
<> object ReloadBtn:
TButton
Left = 555
Top = 64
Width = 78
Height = 25
Caption = 'Reload'
TabOrder = 7
OnClick = ReloadBtnClick
end
<> object GroupBox2:
TGroupBox
Left = 136
Top = 8
Width = 129
Height = 41
Caption = 'Text Color'
TabOrder = 8
<> object RadioButton3:
TRadioButton
Left = 8
Top = 16
Width = 57
Height = 17
Caption = 'Black'
Checked = True
Font.Color = clBlack
Font.Height = 13

UnderlineLink = False
OnHighlightText =
TextBrowserHighlightText
OnTextChanged =
TextBrowserTextChanged
end
<> object GroupBox1:
TGroupBox
Left = 8
Top = 8
Width = 121
Height = 41
Caption = 'Link Color'
TabOrder = 3
<> object RadioButton1:
TRadioButton
Left = 8
Top = 16 Width = 57
Height = 17
Caption = 'Blue'
Checked = True
Font.Color = clBlue
Font.Height = 13
Font.Name = 'Helvetica'
Font.Pitch = fpVariable
Font.Style = [fsBold]
ParentFont = False
TabOrder = 0
OnClick = RadioButton1Click
end
<> object RadioButton2:
TRadioButton
Left = 64
<b>Top</b> = 17
Width = 49
Height = 17
Caption = 'Red'
Font.Color = clRed
Font.Height = 13
Font.Name = 'Helvetica'
Font.Pitch = fpVariable
Font.Style = [fsBold]
ParentFont = False
TabOrder = 1
TabStop = False
OnClick = RadioButton2Click
end

OnClick =
UnderlineCheckboxClick
end
<> object StatusBar:
TStatusBar
Left = 0
Top = 450
Width = 640
Height = 19
Panels = <>
SimplePanel = True
SizeGrip = False
end
End

Font.Name = 'Helvetica'
Font.Pitch = fpVariable
Font.Style = [fsBold]
ParentFont = False
TabOrder = 0
OnClick = RadioButton3Click
end
<> object RadioButton4:
TRadioButton
Left = 65
Top = 16
Width = 58
Height = 17
Caption = 'Green'
Font.Color = clGreen
Font.Height = 13
Font.Name = 'Helvetica'
Font.Pitch = fpVariable
Font.Style = [fsBold]
ParentFont = False
TabOrder = 1
TabStop = False
OnClick = RadioButton4Click
end
end
<> object RadioGroup1:
TRadioGroup
Left = 264
Top = 48
Width = 161
Width = 161 Height = 52
Width = 161 Height = 52 Items.Strings = (
Width = 161 Height = 52 Items.Strings = ( 'Formatted'
Width = 161 Height = 52 Items.Strings = ( 'Formatted' 'Plain')
Width = 161 Height = 52 Items.Strings = ( 'Formatted' 'Plain') Caption = 'Text Format'
Width = 161 Height = 52 Items.Strings = ( 'Formatted' 'Plain') Caption = 'Text Format' ItemIndex = 0
Width = 161 Height = 52 Items.Strings = ( 'Formatted' 'Plain') Caption = 'Text Format' ItemIndex = 0 TabOrder = 9
Width = 161 Height = 52 Items.Strings = ( 'Formatted' 'Plain') Caption = 'Text Format' ItemIndex = 0 TabOrder = 9 OnClick = RadioGroup1Click
Width = 161 Height = 52 Items.Strings = ( 'Formatted' 'Plain') Caption = 'Text Format' ItemIndex = 0 TabOrder = 9 OnClick = RadioGroup1Click Orientation = orHorizontal
Width = 161 Height = 52 Items.Strings = ( 'Formatted' 'Plain') Caption = 'Text Format' ItemIndex = 0 TabOrder = 9 OnClick = RadioGroup1Click Orientation = orHorizontal end
Width = 161 Height = 52 Items.Strings = ( 'Formatted' 'Plain') Caption = 'Text Format' ItemIndex = 0 TabOrder = 9 OnClick = RadioGroup1Click Orientation = orHorizontal end <> object
Width = 161 Height = 52 Items.Strings = ( 'Formatted' 'Plain') Caption = 'Text Format' ItemIndex = 0 TabOrder = 9 OnClick = RadioGroup1Click Orientation = orHorizontal end <> object UnderlineCheckbox: TCheckBox
Width = 161 Height = 52 Items.Strings = ( 'Formatted' 'Plain') Caption = 'Text Format' ItemIndex = 0 TabOrder = 9 OnClick = RadioGroup1Click Orientation = orHorizontal end <> object UnderlineCheckbox: TCheckBox Left = 432
Width = 161 Height = 52 Items.Strings = ( 'Formatted' 'Plain') Caption = 'Text Format' ItemIndex = 0 TabOrder = 9 OnClick = RadioGroup1Click Orientation = orHorizontal end <> object UnderlineCheckbox: TCheckBox Left = 432 Top = 60
Width = 161 Height = 52 Items.Strings = ( 'Formatted' 'Plain') Caption = 'Text Format' ItemIndex = 0 TabOrder = 9 OnClick = RadioGroup1Click Orientation = orHorizontal end <> object UnderlineCheckbox: TCheckBox Left = 432 Top = 60 Width = 120
Width = 161 Height = 52 Items.Strings = ( 'Formatted' 'Plain') Caption = 'Text Format' ItemIndex = 0 TabOrder = 9 OnClick = RadioGroup1Click Orientation = orHorizontal end <> object UnderlineCheckbox: TCheckBox Left = 432 Top = 60 Width = 120 Height = 35
Width = 161 Height = 52 Items.Strings = ( 'Formatted' 'Plain') Caption = 'Text Format' ItemIndex = 0 TabOrder = 9 OnClick = RadioGroup1Click Orientation = orHorizontal end <> object UnderlineCheckbox: TCheckBox Left = 432 Top = 60 Width = 120

<u> </u>	
private	
{ Private declarations }	
public	
{ Public declarations }	مج ا
end;	
<>procedure ShowAboutBox;	
implementation	
•	
{\$R *.xfm}	•
,	program
<>procedure ShowAboutBox;	program
begin	uses
with	
TAboutForm.Create(Application)	QForms MainFr
do	MainFo
begin	AboutF
ShowModal;	AboutFo
Free;	EditFrn
end;	
end;	{EditFor
	1
<>procedure	begin
TAboutForm.Button1Click(Sender:	A 1: 4:
TObject);	Applicati
begin	m, Mainl
Close;	Applica
end;	end.
end.	111111
chu.	//////
\\\\\	•, 4 %
	unit Abo
	• • •
unit EditFrm;	interface
unit Editt ini,	
interface	uses
interface	SysUtils
NGOS	QGraphi
Systicia Types Classes	QDialogs
SysUtils, Types, Classes,	QStdCt
QGraphics, QControls, QForms,	
QDialogs,	type
QStdCtrls;	TAbout
	Button
type	Label1
TEditForm = class(TForm)	<
memText: TMemo;	Button10

Procedure\$22
الملفالتنفيذي لبرنامج
(BasicEd)
إعداد
علاء الدين اللباد
واتف ۷۱۳۵۷۵۶۶۹۰
program BasicEd;
OF O
QForms,  MainFrm in 'MainFrm.pas'
{MainForm},
AboutFrm in 'AboutFrm.pas'
{AboutForm},
EditFrm in 'EditFrm.pas'
{EditForm};
begin
Application.CreateForm(TMainForm, MainForm);
Application.Run;
end.
\\\\\\
unit AboutFrm;
interface
USOS
SysUtils, Types, Classes,
QGraphics, QControls, QForms,
QDialogs,
QStdCtrls;
tyne
type TAboutForm = class(TForm)
Button1: TButton;
Label1: TLabel;
<>procedure
Button1Click(Sender: TObject);

ToolButton5: TToolButton;
ToolButton6: TToolButton;
mnuFile: TMenuItem;
mnuNew: TMenuItem;
mnuOpen: TMenuItem;
mnuSave: TMenuItem;
mnuEdit: TMenuItem;
mnuCut: TMenuItem;
mnuCopy: TMenuItem;
mnuPaste: TMenuItem;
mnuExit: TMenuItem;
mnuHelp: TMenuItem;
mnuAbout: TMenuItem;
OpenDialog: TOpenDialog;
SaveDialog: TSaveDialog;
ActionList: TActionList;
actnCut: TEditCut;
actnCopy: TEditCopy;
actnPaste: TEditPaste;
actnSelectAll: TEditSelectAll;
actnDelete: TEditDelete;
actnWinClose: TWindowClose;
actnWinCascade:
TWindowCascade;
actnWinTile: TWindowTile;
actnWinMinimizeAll:
TWindowMinimizeAll;
actnNew: TAction;
actnSave: TAction;
actnOpen: TAction;
actnExit: TAction;
mnuWindow: TMenuItem;
mnuTile: TMenuItem;
N1: TMenuItem;
mnuSelectAll: TMenuItem;
mnuMinimize: TMenuItem;
mnuCascade: TMenuItem;
mnuClose: TMenuItem;
N2: TMenuItem;
N3: TMenuItem;
ToolButton7: TToolButton;
ImageList: TImageList;
<>procedure
mnuAboutClick(Sender: TObject);
<>procedure
actnNewExecute(Sender: TObject);
<>procedure
actnSaveExecute(Sender: TObject);
acingaveraceute(senuel. 100ject);

<>procedure
FormClose(Sender: TObject; var
Action: TCloseAction);
private
{ Private declarations }
public
{ Public declarations }
end;
implementation
{\$R *.xfm}
<>procedure
TEditForm.FormClose(Sender:
TObject; var Action: TCloseAction);
begin
Action := caFree;
end;
end.
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
unit MainFrm;
interface
uses
SysUtils, Types, Classes,
QGraphics, QControls, QForms,
QDialogs,
QExtCtrls, QComCtrls, QTypes,
QMenus, QStdCtrls, QImgList,
QActnList,
QStdActns;
type
TMainForm = class(TForm)
MainMenu: TMainMenu;
StatusBar1: TStatusBar;
ToolBar1: TToolBar;
ToolBar1: TToolBar; ToolButton1: TToolButton;
ToolBar1: TToolBar; ToolButton1: TToolButton; ToolButton2: TToolButton;
ToolBar1: TToolBar; ToolButton1: TToolButton;

begin ShowAboutBox; end; <>procedure TMainForm.actnNewExecute(Sende	<>p
end; <>procedure	<>p
<>procedure	-
-	acts ExitE-
-	actnExitExe
TMainForm.actnNewExecute(Sende	<>p
	actnWinClo
r: TObject);	TObject);
begin	<>p
CreateMDIChild(sUntitled +	FormCreate
IntToStr(MDIChildCount + 1));	private
end;	<>p
	CreateMDI
<>procedure	string);
TMainForm.actnSaveExecute(Sende	{ Private
r: TObject);	public
begin	{ Public d
if not Assigned(ActiveMDIChild)	end;
then	
Exit;	var
SaveDialog.FileName :=	MainForm
ActiveMDIChild.Caption;	
if SaveDialog.Execute then	implementa
begin	
if ActiveMDIChild is TEditForm	{\$R *.xfm}
then	(WIX AIII)
then	uses
TEditForm(ActiveMDIChild).memT	AboutFrm
ext.Lines.SaveToFile(SaveDialog.Fil	Aboutifili
eName);	resourcestri
ActiveMDIChild.Caption :=	sUntitled =
SaveDialog.FileName;	sontitied –
_	
end; end;	<>pro
ciiu,	TMainForm
< >nwoooduwo	Name: strin
<>procedure	var
TMainForm.actnOpenExecute(Send	Child: TE
er: TObject);	begin
begin	{ create a r
if OpenDialog.Execute then	Child :=
begin	TEditForm.
if ActiveMDIChild is TEditForm	Child.Cap
then	if FileExist
	Child.Mem
TEditForm(ActiveMDIChild).memT	e(Name);
ext.Lines.LoadFromFile(OpenDialog	end;
.FileName);	
ActiveMDIChild.Caption :=	<>pro
<del>-</del>	
OpenDialog.FileName; end;	TMainForm

<>procedure
actnOpenExecute(Sender: TObject);
<pre>&lt;&gt;procedure actnExitExecute(Sender: TObject);</pre>
<>procedure
actnWinCloseExecute(Sender:
TObject);
<>procedure
FormCreate(Sender: TObject);
private
<>procedure
CreateMDIChild(const Name:
string);
{ Private declarations }
public
{ Public declarations }
end;
var
MainForm: TMainForm;
• • •
implementation
{\$R *.xfm}
uses E Pag
AboutFrm, EditFrm;
resourcestring
sUntitled = 'Untitled';
sondied – Ondied,
<>procedure
TMainForm.CreateMDIChild(const
Name: string);
var
Child: TEditForm;
begin
{ create a new MDI child window }
Child :=
TEditForm.Create(Application);
Child.Caption := Name;
if FileExists(Name) then
Child.MemText.Lines.LoadFromFil
e(Name);
end;
<>procedure
TMainForm.mnuAboutClick(Sender
: TObject);

Position = poDefault
Scaled = False
OnCreate = FormCreate
PixelsPerInch = 96
<>Object StatusBar1:
TStatusBar
Left = 0
Top = 339
Width = 480
Height = 19
Font.CharSet = fcsLatin1
Font.Color = clBlack
Font.Height = 13
Font.Name = 'Helvetica'
Font.Pitch = fpVariable
Font.Style = []
Font.Weight = 40
Panels = <>
ParentColor = True
ParentFont = False
end
<>Object ToolBar1:
TToolBar
Left = 0
Top = 0
Width = 480
Height = 29
ButtonHeight = 23
Flat = True
Images = ImageList
TabOrder = 1
<>Object ToolButton1:
TToolButton
HelpType = htContext
Left = 1
Top = 4
Height = 23
Action = actnNew
Caption = '&New' ParentShowHint = False
ShowHint = True
end
<>Object ToolButton2:
TToolButton
HelpType = htContext
Left = 47
Top = 4
Height = 23

end;
<>procedure
TMainForm.actnExitExecute(Sende
r: TObject);
begin
Close;
end;
· · · · · · · · · · · · · · · · · · ·
<>procedure
TMainForm.actnWinCloseExecute(
Sender: TObject);
begin
ActiveMDIChild.Close;
end;
,
<>procedure
TMainForm.FormCreate(Sender:
TObject);
begin
actnNewExecute(nil);
end;
end.
\\\\\

	<>Object\$54		
	الملف النصي لبرناهم		
	(BasicEd)		
عامدا			
	علاء الدين اللباد		
	هاتف ۷۳۷۱۹۹۱		
<>Object MainForm:			
1	MainForm		
	Left = 183		
	Top = 108		
	Width = 480		
	Height = 381		
	VertScrollBar.Range = 48		
Caption = 'Basic Edit'			
	Color = clBackground		
	FormStyle = fsMDIForm		
	Menu = MainMenu		

end		
<>Object ToolButton7:		
TToolButton		
HelpType = htContext		
Left = 24		
Top = 4		
Height = 23		
Action = actnOpen		
Caption = '&Open'		
end		
end		
<>Object MainMenu:		
TMainMenu		
Images = ImageList		
Left = 24		
Top = 40 <>Object mnuFile:		
TMenuItem		
Caption = 'File'		
HelpType = htContext		
<>Object mnuNew:		
TMenuItem		
Action = actnNew		
end		
<>Object mnuOpen:		
TMenuItem		
Action = actnOpen		
end		
<>Object mnuSave:		
TMenuItem		
Action = actnSave		
end		
<>Object N3: TMenuItem		
Caption = '-'		
HelpType = htContext		
end		
<>Object mnuExit:		
TMenuItem		
Action = actnExit		
end		
end end		
<>Object mnuEdit:		
TMenuItem		
Caption = 'Edit'		
HelpType = htContext		
<>Object mnuCut:		
TMenuItem		
Action = actnCut		

Action = actnSave
Caption = 'Save As'
ParentShowHint = False
ShowHint = True
end
<>Object ToolButton3:
TToolButton
Left = 70
Top = 4
Width = 8
Height = 23
Style = tbsSeparator
Caption = 'ToolButton3'
ImageIndex = 2
Wrap = True
end
<>Object ToolButton4:
TToolButton
HelpType = htContext
Left = 78
<b>Top</b> = <b>4</b>
Height = 23
Action = actnCut
Caption = 'Cu&t'
ParentShowHint = False
ShowHint = True
end
<>Object ToolButton5:
TToolButton
HelpType = htContext
Left = 101
Top = 4
Height = 23
Action = actnCopy
Caption = '&Copy'
ParentShowHint = False
ShowHint = True
end
<>Object ToolButton6:
TToolButton
HelpType = htContext
Left = 124
Top = 4
Height = 23
Action = actnPaste
Caption = '&Paste'
ParentShowHint = False
ShowHint = True
Showing - 11 ac

Caption = 'About' OnClick = mnuAboutClick HelpType = htContext end end end <>Object OpenDialog: TOpenDialog Filter = '*.txt *.*'#10 FilterIndex = 0
OnClick = mnuAboutClick HelpType = htContext end end end <>Object OpenDialog: TOpenDialog Filter = '*.txt *.*'#10
end end end <>Object OpenDialog: TOpenDialog Filter = '*.txt *.*'#10
end end end <>Object OpenDialog: TOpenDialog Filter = '*.txt *.*'#10
end <>Object OpenDialog: TOpenDialog Filter = '*.txt *.*'#10
<>Object OpenDialog: TOpenDialog Filter = '*.txt *.*'#10
TOpenDialog Filter = '*.txt *.*'#10
Filter = '*.txt *.*'#10
FilterIndex = 0
Height = 0
Title = 'Open a text file'
Width = 0
Left = 168
Top = 40
end
<>Object SaveDialog:
TSaveDialog
DefaultExt = 'txt'
FilterIndex = 0
Height = 0
Title = 'Save As'
Width = 0
Left = 96
Top = 40
end
<>Object ActionList:
TActionList
Images = ImageList
Left = 240
Top = 40
<>Object actnNew: TAction
Category = 'File'
Caption = '&New'
Hint = 'Create a new text file'
ImageIndex = 0
OnExecute = actnNewExecute
end
<>Object actnCut: TEditCut
Category = 'Edit'
Caption = 'Cu&t'
Hint = 'Cut'
ImageIndex = 3
ShortCut = 16472
end
<>Object actnCopy:
TEditCopy

_		
end		
<>Object mnuCopy:		
TMenuItem		
Action = actnCopy		
end		
<>Object mnuPaste:		
TMenuItem		
Action = actnPaste		
end		
<>Object N1: TMenuItem Caption = '-'		
*		
HelpType = htContext		
end		
<>Object mnuSelectAll:		
TMenuItem		
Action = actnSelectAll		
end		
end		
<>Object mnuWindow:		
TMenuItem		
Caption = 'Window'		
HelpType = htContext		
<>Object mnuTile:		
TMenuItem		
Action = actnWinTile		
end		
<>Object mnuMinimize:		
TMenuItem		
Action = actnWinMinimizeAll		
end		
<>Object mnuCascade:		
TMenuItem		
Action = actnWinCascade		
end		
<>Object N2: TMenuItem		
Caption = '-'		
HelpType = htContext		
end		
<>Object mnuClose:		
TMenuItem		
Action = actnWinClose		
end		
end		
<>Object mnuHelp:		
TMenuItem		
Caption = 'Help'		
HelpType = htContext		
<>Object mnuAbout:		

Catagory - Window!		
Category = 'Window'		
Caption = '&Tile Windows'		
end		
<>Object actnWinMinimizeAll:		
TWindowMinimizeAll		
Category = 'Window'		
Caption = '&Minimize All'		
end		
<>Object actnSave: TAction		
Category = 'File'		
Caption = 'Save As'		
Hint = 'Save the current file as'		
ImageIndex = 2		
ShortCut = 16467		
OnExecute = actnSaveExecute		
end		
<>Object actnExit: TAction		
Category = 'File'		
Caption = 'E&xit'		
OnExecute = actnExitExecute		
end		
end		
<>Object ImageList:		
TImageList		
Left = 304		
Top = 40		
Bitmap = {		
end		
End		
\\\\		
<>Object AboutForm:		
TAboutForm		
Left = 164		
Top = 119		
Width = 365		
Height = 220		
VertScrollBar.Range = 193		
HorzScrollBar.Range = 353		
ActiveControl = Button1		
BorderStyle = fbsDialog		
Caption = 'About'		
Color = clBackground		
Constraints.MaxHeight = 220		
Constraints.MaxWidth = 365		
Constraints.MinHeight = 220		
Constraints.MinWidth = 365		
Consti amits.ivim vy iutii – 303		

Category = 'Edit'
Caption = '&Copy'
Hint = 'Copy'
ImageIndex = 4
ShortCut = 16451
end
<>Object actnOpen:
TAction
Category = 'File'
Caption = '&Open'
Hint = 'Open a text file'
ImageIndex = 1
OnExecute = actnOpenExecute
end
<>Object actnPaste:
TEditPaste
Category = 'Edit'
Caption = '&Paste'
Hint = 'Paste'
ImageIndex = 5
ShortCut = 16470
end
<>Object actnSelectAll:
TEditSelectAll
Category = 'Edit'
Caption = 'Select &All'
end
<>Object actnDelete:
TEditDelete
Category = 'Edit'
Caption = '&Delete'
ImageIndex = 6
ShortCut = 4103
end
<>Object actnWinClose:
TWindowClose
Category = 'Window'
Caption = 'C&lose'
OnExecute =
actnWinCloseExecute
end
<>Object actnWinCascade:
TWindowCascade
Category = 'Window'
Caption = '&Cascade'
end
<>Object actnWinTile:
TWindowTile
1 WILLIAM LIIG

TabOrder = 0
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	Top = 168
	Width = 75
Procedure\$29	Height = 25
	Caption = 'OK'
	TabOrder = 1
الملف التنفيذي لبرنامج	OnClick = Butto
(docking)	end
(docking)	<>Object La
	Left = 16
1 111 11 4	Top = 16
علاء الدين اللباد	Width = 337
هاتف ۷۳۷۷۷۶۹۰	Height = 30
	AutoSize = False
unit uMain;	Caption = 'Simpl 2001 Borland Softw
	Layout = tlCente
interface	end
	End
uses	
Windows, Messages, SysUtils,	
Classes, Graphics, Controls, Forms, Dialogs,	
Menus, StdCtrls, ComCtrls,	\\\\\
ActnList, ToolWin, ExtCtrls,	
uDockForm;	<>Object Edi
	TEditForm
type	<b>Left = 162</b>
TMainForm = class(TForm)	Top = 119
CoolBar1: TCoolBar;	<b>Width = 357</b>
ToolBar1: TToolBar;	<b>Height = 267</b>
ToolBar2: TToolBar;	ActiveControl = m
ToolButton1: TToolButton;	Caption = 'EditFo
ToolButton2: TToolButton;	Color = clBackgro
ToolButton3: TToolButton;	FormStyle = fsMI
ToolButton5: TToolButton;	Position = poDefar
ToolButton6: TToolButton;	OnClose = FormC PixelsPerInch = 75
ToolButton7: TToolButton;	TextHeight = 13
ToolButton13: TToolButton;	TextWidth = 6
btnToolBar1: TToolButton;	<>Object me
btnToolBar2: TToolButton;	Left = 0
ActionList1: TActionList;	Top = 0
ViewToolBar1: TAction; ViewToolBar2: TAction;	Width = 357
LeftDockPanel: TPanel;	<b>Height = 267</b>
BottomDockPanel: TPanel;	Align = alClient
DUMUMDUCKI AMEI. 11 AMEI,	<b>B</b>

Scaled = False
Scalca Paist
PixelsPerInch = 96
<>Object Button1: TButton
Left = 144
Top = 168
Width = 75
Height = 25
Caption = 'OK'
TabOrder = 1
OnClick = Button1Click
end
<>Object Label1: TLabel
Left = 16
Top = 16
Width = 337
Height = 30
AutoSize = False
Caption = 'Simple Text Editor (c)
2001 Borland Software Corporation'
Layout = tlCenter
end
End
\\\\\
<>Object EditForm:
TEditForm
Left = 162
Top = 119
Width = 357
Height = 267
ActiveControl = memText
ActiveControl = memText
ActiveControl = memText Caption = 'EditForm'
ActiveControl = memText  Caption = 'EditForm'  Color = clBackground
ActiveControl = memText  Caption = 'EditForm'  Color = clBackground  FormStyle = fsMDIChild
ActiveControl = memText  Caption = 'EditForm'  Color = clBackground  FormStyle = fsMDIChild  Position = poDefault
ActiveControl = memText  Caption = 'EditForm'  Color = clBackground  FormStyle = fsMDIChild  Position = poDefault  OnClose = FormClose
ActiveControl = memText Caption = 'EditForm' Color = clBackground FormStyle = fsMDIChild Position = poDefault OnClose = FormClose PixelsPerInch = 75
ActiveControl = memText  Caption = 'EditForm'  Color = clBackground  FormStyle = fsMDIChild  Position = poDefault  OnClose = FormClose  PixelsPerInch = 75  TextHeight = 13
ActiveControl = memText  Caption = 'EditForm'  Color = clBackground  FormStyle = fsMDIChild  Position = poDefault  OnClose = FormClose  PixelsPerInch = 75  TextHeight = 13  TextWidth = 6
ActiveControl = memText Caption = 'EditForm' Color = clBackground FormStyle = fsMDIChild Position = poDefault OnClose = FormClose PixelsPerInch = 75 TextHeight = 13 TextWidth = 6 <>Object memText: TMemo
ActiveControl = memText  Caption = 'EditForm'  Color = clBackground  FormStyle = fsMDIChild  Position = poDefault  OnClose = FormClose  PixelsPerInch = 75  TextHeight = 13  TextWidth = 6  <>Object memText: TMemo  Left = 0
ActiveControl = memText  Caption = 'EditForm'  Color = clBackground  FormStyle = fsMDIChild  Position = poDefault  OnClose = FormClose  PixelsPerInch = 75  TextHeight = 13  TextWidth = 6  <>Object memText: TMemo  Left = 0  Top = 0
ActiveControl = memText  Caption = 'EditForm'  Color = clBackground  FormStyle = fsMDIChild  Position = poDefault  OnClose = FormClose  PixelsPerInch = 75  TextHeight = 13  TextWidth = 6  <>Object memText: TMemo  Left = 0  Top = 0  Width = 357
ActiveControl = memText  Caption = 'EditForm'  Color = clBackground  FormStyle = fsMDIChild  Position = poDefault  OnClose = FormClose  PixelsPerInch = 75  TextHeight = 13  TextWidth = 6  <>Object memText: TMemo  Left = 0  Top = 0

TOL	VO PH TEC PH
TObject;	VSplitter: TSplitter;
Source: TDragDockObject; X,	HSplitter: TSplitter;
Y: Integer; State: TDragState;	MainMenu1: TMainMenu;
var Accept: Boolean);	File2: TMenuItem;
<> procedure	Exit2: TMenuItem;
BottomDockPanelDockOver(Sender	View2: TMenuItem;
: TObject;	N2: TMenuItem;
Source: TDragDockObject; X,	ToolBar21: TMenuItem;
Y: Integer; State: TDragState;	ToolBar11: TMenuItem;
var Accept: Boolean);	ToolButton16: TToolButton;
<> procedure	ViewWhiteWindow: TAction;
LeftDockPanelUnDock(Sender:	ExitAction: TAction;
TObject; Client: TControl;	ViewBlueWindow: TAction;
NewTarget: TWinControl; var	ViewGreenWindow: TAction;
Allow: Boolean);	ViewRedWindow: TAction;
<> procedure	,
ExitActionExecute(Sender:	ViewTealWindow: TAction;
TObject);	ViewPurpleWindow: TAction;
<> procedure	ViewLimeWindow: TAction;
ViewWhiteWindowExecute(Sender:	ToolButton4: TToolButton;
TObject);	White1: TMenuItem;
	Blue1: TMenuItem;
<> procedure	Green1: TMenuItem;
LeftDockPanelGetSiteInfo(Sender:	Lime1: TMenuItem;
TObject;	Purple1: TMenuItem;
DockClient: TControl; var	Red1: TMenuItem;
InfluenceRect: TRect; MousePos:	Teal1: TMenuItem;
TPoint;	<> procedure
var CanDock: Boolean);	FormCreate(Sender: TObject);
private	<> procedure
<> procedure	CoolBar1DockOver(Sender:
CreateDockableWindows;	TObject; Source:
public	TDragDockObject; X,
<> procedure	Y: Integer; State: TDragState;
ShowDockPanel(APanel: TPanel;	var Accept: Boolean);
MakeVisible: Boolean; Client:	· /
TControl);	<pre>&lt;&gt; procedure ViowToolPar1Evacuta(Sandar)</pre>
end;	ViewToolBar1Execute(Sender:
	TObject);
var	<pre>&lt;&gt; procedure ViewTeelDer2Everyte(Sender)</pre>
MainForm: TMainForm;	ViewToolBar2Execute(Sender:
	TObject);
implementation	<> procedure
mpiementation	Exit1Click(Sender: TObject);
uses uTabHost, uConjoinHost;	<> procedure
uses u l'abilost, uconjoninost;	LeftDockPanelDockDrop(Sender:
(CD * df)	TObject;
{\$R *.dfm}	Source: TDragDockObject; X,
	Y: Integer);
const	<> procedure
Colors: array[06] of TColor =	LeftDockPanelDockOver(Sender:

nder: TObject; Source:	(clWhite, clBlue, clGreen, clRed,
TDragDockObject;	clTeal,
X, Y: Integer; State: TDragState;	clPurple,
var Accept: Boolean);	clLime);
var	ColStr: array[06] of string =
ARect: TRect;	('White', 'Blue', 'Green', 'Red',
begin	'Teal',
Accept := (Source.Control is	'Purple',
TToolBar);	'Lime');
if Accept then	
begin	var
//Modify the DockRect to preview	DockWindows: array[06] of
dock area (Coolbar client area)	TDockableForm;
ARect.TopLeft :=	
CoolBar1.ClientToScreen(CoolBar1.	{TMainForm}
ClientRect.TopLeft);	
ARect.BottomRight :=	procedure
CoolBar1.ClientToScreen(CoolBar1.	TMainForm.FormCreate(Sender:
ClientRect.BottomRight);	TObject);
Source.DockRect := ARect;	begin
end;	CreateDockableWindows;
end;	end;
ciu,	enu,
nuacadana	nucacduus
procedure	procedure TMainForm.CreateDockableWindo
TMainForm.ViewToolBar1Execute(	
Sender: TObject);	ws;
begin	var
//Toggles the visible state of	I: Integer;
Toolbar1, regardless of it's docked	begin
state.	for I := 0 to High(DockWindows)
ToolBar11.Checked := not	do
ToolBar11.Checked;	begin
btnToolBar1.Down :=	DockWindows[I] :=
ToolBar11.Checked;	TDockableForm.Create(Application
if ToolBar1.Floating then	);
ToolBar1.HostDockSite.Visible :=	DockWindows[I].Caption :=
ToolBar11.Checked	ColStr[I];
else	DockWindows[I].Memo1.Color :=
ToolBar1.Visible :=	Colors[I];
ToolBar11.Checked;	
end;	DockWindows[I].Memo1.Font.Color
	:= Colors[I] xor \$00FFFFFF;
procedure	DockWindows[I].Memo1.Text :=
TMainForm.ViewToolBar2Execute(	ColStr[I] + ' window ';
Sender: TObject);	end;
begin	end;
//Toggles the visible state of	,
Toolbar2, regardless of it's docked	procedure
state.	TMainForm.CoolBar1DockOver(Se
suit.	1 Maint of in-Cooldal IDUCKOVEI (SC

A.D ED	T ID 44 CL 1 1
ARect: TRect;	ToolBar21.Checked := not
begin	ToolBar21.Checked;
Accept := Source.Control is	btnToolBar2.Down :=
TDockableForm;	ToolBar21.Checked;
if Accept then	if ToolBar2.Floating then
begin	
//Modify the DockRect to preview	TToolDockForm(ToolBar2.HostDoc
dock area.	kSite).Visible := ToolBar21.Checked
ARect.TopLeft :=	else
LeftDockPanel.ClientToScreen(Poin	ToolBar2.Visible :=
t(0,0));	ToolBar21.Checked;
ARect.BottomRight :=	end;
LeftDockPanel.ClientToScreen(	
Point(Self.ClientWidth div 3,	procedure
LeftDockPanel.Height));	TMainForm.Exit1Click(Sender:
Source.DockRect := ARect;	TObject);
end;	begin
end;	Close;
	end;
procedure	
TMainForm.BottomDockPanelDock	procedure
Over(Sender: TObject;	TMainForm.LeftDockPanelDockDr
Source: TDragDockObject; X, Y:	op(Sender: TObject;
Integer; State: TDragState;	Source: TDragDockObject; X, Y:
var Accept: Boolean);	Integer);
var	begin
ARect: TRect;	//OnDockDrop gets called AFTER
begin	the client has actually docked,
Accept := Source.Control is	//so we check for DockClientCount
TDockableForm;	= 1 before making the dock panel
if Accept then	visible.
begin	if (Sender as
//Modify the DockRect to preview	TPanel).DockClientCount = 1 then
dock area.	ShowDockPanel(Sender as
ARect.TopLeft :=	TPanel, True, nil);
BottomDockPanel.ClientToScreen(	(Sender as
Point(0, -Self.ClientHeight div	TPanel).DockManager.ResetBounds
3));	(True);
ARect.BottomRight :=	//Make DockManager repaints it's
BottomDockPanel.ClientToScreen(	clients.
Point(BottomDockPanel.Width,	end;
BottomDockPanel.Height));	
Source.DockRect := ARect;	procedure
end;	TMainForm.LeftDockPanelDockOv
·	er(Sender: TObject;
end;	Source: TDragDockObject; X, Y:
nwaaduwa	Integer; State: TDragState;
procedure TMoinForm LoftDookBonolUnDook(	var Accept: Boolean);
TMainForm.LeftDockPanelUnDock(	* //
Sender: TObject; Client: TControl;	var

ICAR I TAR IR IA	NI TO CONTRACT OF A STATE OF THE STATE OF TH
if APanel = LeftDockPanel then	NewTarget: TWinControl; var
begin	Allow: Boolean);
APanel.Width := ClientWidth	begin
div 3;	//OnUnDock gets called BEFORE
VSplitter.Left := APanel.Width	the client is undocked, in order to
+ VSplitter.Width;	optionally
end	//disallow the undock.
else begin	DockClientCount is never 0 when
APanel.Height := ClientHeight	called from this event.
div 3;	if (Sender as
HSplitter.Top := ClientHeight -	TPanel).DockClientCount = 1 then
APanel.Height - HSplitter.Width;	ShowDockPanel(Sender as
end	TPanel, False, nil);
else	end;
if APanel = LeftDockPanel then	
APanel.Width := 0	procedure
else	TMainForm.ShowDockPanel(APane
APanel.Height := 0;	l: TPanel; MakeVisible: Boolean;
if MakeVisible and (Client <> nil)	Client: TControl);
then Client.Show;	begin
end;	//Client - the docked client to show
	if we are re-showing the panel.
procedure	//Client is ignored if hiding the
TMainForm.ExitActionExecute(Sen	panel.
der: TObject);	
begin	//Since docking to a non-visible
Close;	docksite isn't allowed, instead of
end;	setting
	//Visible for the panels we set the
procedure	width to zero. The default
TMainForm.ViewWhiteWindowExe	InfluenceRect
cute(Sender: TObject);	//for a control extends a few pixels
var	beyond it's boundaries, so it is
DockWindow: TDockableForm;	possible
begin	//to dock to zero width controls.
DockWindow :=	100
DockWindows[(Sender as	//Don't try to hide a panel which
TComponent). Tag];	has visible dock clients.
with DockWindow do	if not MakeVisible and
//if the docked window is	(APanel.VisibleDockClientCount >
TabDocked, it is docked to the	1) then
PageControl	Exit;
//(owned by TTabDockHost) so	
show the host form.	if APanel = LeftDockPanel then
if HostDockSite is TPageControl	VSplitter.Visible := MakeVisible
then	else
	HSplitter.Visible := MakeVisible;
TTabDockHost(HostDockSite.Owne	
r).Show	if MakeVisible then

unit uConjoinHost;	else
unit uconjoinitiost,	//If window is conjoin-docked,
interface	host and/or form may not be visible
Interface	//so show both.
77.00	
uses	if (HostDockSite is
Windows, Messages, SysUtils,	TConjoinDockHost) and not
Classes, Graphics, Controls, Forms,	HostDockSite.Visible then
Dialogs, uDockForm;	begin
	HostDockSite.Show;
type	
TConjoinDockHost = class(TForm)	TConjoinDockHost(HostDockSite).U
<> procedure	pdateCaption(nil);
FormClose(Sender: TObject; var	DockWindow.Show;
Action: TCloseAction);	end else
<> procedure	//If form is docked to one of the
FormDockDrop(Sender: TObject;	"hidden" docking panels, resize the
Source: TDragDockObject; X,	//panel and re-show the docked
Y: Integer);	form.
<> procedure	if (HostDockSite is TPanel)
FormUnDock(Sender: TObject;	and ((HostDockSite.Height = 0)
Client: TControl;	or (HostDockSite.Width = 0)) then
NewTarget: TWinControl; var	or (Hoseb ornard) which
Allow: Boolean);	MainForm.ShowDockPanel(HostDo
<> procedure	ckSite as TPanel, True,
FormDockOver(Sender: TObject;	DockWindow)
Source: TDragDockObject; X,	else
Y: Integer; State: TDragState;	//if the window isn't docked at
var Accept: Boolean);	all, simply show it.
<pre>&lt;&gt; procedure Form CotSite Info(Sondon TObicot)</pre>	DockWindow.Show;
FormGetSiteInfo(Sender: TObject;	end;
DockClient: TControl;	
var InfluenceRect: TRect;	procedure
MousePos: TPoint; var CanDock:	TMainForm.LeftDockPanelGetSiteI
Boolean);	nfo(Sender: TObject;
private	DockClient: TControl; var
<> procedure	InfluenceRect: TRect; MousePos:
DoFloat(AControl: TControl);	TPoint;
public	var CanDock: Boolean);
<> procedure	begin
<b>UpdateCaption(Exclude: TControl)</b> ;	//if CanDock is true, the panel will
end;	not automatically draw the preview
	rect.
var	CanDock := DockClient is
ConjoinDockHost:	TDockableForm;
TConjoinDockHost;	end;
<b>y</b>	,
implementation	end.
{\$R *.dfm}	\\\\\\\\\\
γυν ·uiii)	

Cantian - Cantian I	
Caption := Caption +	1
TDockableForm(DockClients[I]).Ca	procedure
ption + ' ';	TConjoinDockHost.DoFloat(AContr
end;	ol: TControl);
	var
procedure	ARect: TRect;
TConjoinDockHost.FormDockDrop(	begin
Sender: TObject;	//float the control with its original
Source: TDragDockObject; X, Y:	size.
Integer);	ARect.TopLeft :=
begin	AControl.ClientToScreen(Point(0,
UpdateCaption(nil);	0));
DockManager.ResetBounds(True);	ARect.BottomRight :=
//Force DockManager to redraw it's	AControl.ClientToScreen(Point(AC
clients.	ontrol.UndockWidth,
end;	,
,	AControl.UndockHeight));
procedure	AControl.ManualFloat(ARect);
TConjoinDockHost.FormUnDock(Se	end;
nder: TObject; Client: TControl;	
NewTarget: TWinControl; var	procedure
Allow: Boolean);	TConjoinDockHost.FormClose(Send
begin	er: TObject;
3	var Action: TCloseAction);
//only 2 dock clients means the host	, ,
must be destroyed and	begin
//the remaining window undocked	if DockClientCount = 1 then
to its old position and size.	begin D. Fill (Ch. 191)
//(Recall that OnUnDock gets called	DoFloat(DockClients[0]);
before the undocking actually	Action := caFree;
occurs)	end else
if Client is TDockableForm then	Action := caHide;
TDockableForm(Client).DockSite	end;
:= True;	
if (DockClientCount = 2) and	procedure
(NewTarget <> Self) then	TConjoinDockHost.UpdateCaption(
PostMessage(Self.Handle,	<b>Exclude: TControl)</b> ;
WM_CLOSE, 0, 0);	var
UpdateCaption(Client);	I: Integer;
end;	begin
	//if a dockable form is undocking, it
procedure	will pass itself in as Exclude
TConjoinDockHost.FormDockOver(	//because even it hasn't actually
Sender: TObject;	been taken out of the DockClient
Source: TDragDockObject; X, Y:	array
Integer; State: TDragState;	//at this point.
var Accept: Boolean);	Caption := ";
begin	for I := 0 to DockClientCount-1 do
Accept := Source.Control is	if DockClients[I].Visible and
TDockableForm;	
1 DUCKADICI UI III,	(DockClients[I] <> Exclude) then

CM_DOCKCLIENT;	end;
public	
end;	procedure
	TConjoinDockHos
	o(Sender: TObject:
implementation	DockClient: TCor
	InfluenceRect: TRe
{\$R *.dfm}	TPoint;
	var CanDock: Boo
uses ComCtrls, uTabHost,	begin
uConjoinHost, uMain;	CanDock := Dock
	TDockableForm;
procedure	end;
TDockableForm.FormDockOver(Se	
nder: TObject;	end.
Source: TDragDockObject; X, Y:	
Integer; State: TDragState;	\\\\\\\\\
var Accept: Boolean);	
var	unit uDockForm;
ARect: TRect;	
begin	interface
Accept := (Source.Control is	
TDockableForm);	uses
//Draw dock preview depending on	Windows, Messag
where the cursor is relative to our	Classes, Graphics,
client area	Dialogs, Menus,
if Accept and	ExtCtrls, StdCtrls
(ComputeDockingRect(ARect,	
$Point(X, Y)) \Leftrightarrow alNone)$ then	type
Source.DockRect := ARect;	TDockableForm =
end;	Memo1: TMemo
	<> procedu
function	FormDockOver(Se
TDockableForm.ComputeDockingR	Source: TDragDoc
ect(var DockRect: TRect;	Y: Integer; Stat
MousePos: TPoint): TAlign;	var Accept: Boolea
var	<> procedu
DockTopRect,	FormClose(Sender
DockLeftRect,	Action: TCloseActi
DockBottomRect,	<> procedu
DockRightRect,	FormShow(Sender
DockCenterRect: TRect;	private
begin	function
Result := alNone;	ComputeDockingR
//divide form up into docking	DockRect: TRect; I
"Zones"	TPoint): TAlign;
DockLeftRect.TopLeft := Point(0,	<> procedur
0);	CMDockClient(var
DockLeftRect.BottomRight :=	TCMDockClient);

end;	
procedure	
TConjoinDockHost.FormGetSiteInf	
o(Sender: TObject;	
DockClient: TControl; var	
InfluenceRect: TRect; MousePos:	
TPoint;	
var CanDock: Boolean);	
begin	
CanDock := DockClient is	
TDockableForm;	
end;	
cnu,	
end.	
\\\\\\\\\\\	
unit uDockForm;	
· · · · · · · · · · · · · · · · · · ·	
interface	
Interface	
uses	
Windows, Messages, SysUtils,	
Classes, Graphics, Controls, Forms,	
Dialogs, Menus,	
ExtCtrls, StdCtrls;	
Externs, Stucture,	
A	
type	
TDockableForm = class(TForm)	
Memo1: TMemo;	
<> procedure	
FormDockOver(Sender: TObject;	
Source: TDragDockObject; X,	
Y: Integer; State: TDragState;	
var Accept: Boolean);	
<> procedure	
FormClose(Sender: TObject; var	
Action: TCloseAction);	
<> procedure	
FormShow(Sender: TObject);	
private	
function	
ComputeDockingRect(var	
DockRect: TRect; MousePos:	
TPoint): TAlign;	
<> procedure	
CMDockClient(var Message:	
Civil och Chene (var iviessage:	
TCMDockClient); message	

ClientHeight div 2;	Point(ClientWidth div 5,
end	ClientHeight);
else	
if PtInRect(DockRightRect,	DockTopRect.TopLeft :=
MousePos) then	Point(ClientWidth div 5, 0);
begin	DockTopRect.BottomRight :=
Result := alRight;	Point(ClientWidth div 5 * 4,
DockRect := DockRightRect;	ClientHeight div 5);
DockRect.Left := ClientWidth	9 //
div 2;	DockRightRect.TopLeft :=
end	Point(ClientWidth div 5 * 4, 0);
else	DockRightRect.BottomRight :=
if PtInRect(DockBottomRect,	Point(ClientWidth, ClientHeight);
MousePos) then	, , , , , , , , , , , , , , , , , , , ,
begin	DockBottomRect.TopLeft :=
Result := alBottom;	Point(ClientWidth div 5,
DockRect :=	ClientHeight div 5 * 4);
DockBottomRect;	DockBottomRect.BottomRight :=
DockRect.Left := 0;	Point(ClientWidth div 5 * 4,
DockRect.Right :=	ClientHeight);
ClientWidth;	V //
DockRect.Top :=	DockCenterRect.TopLeft :=
ClientHeight div 2;	Point(ClientWidth div 5,
end	ClientHeight div 5);
else	DockCenterRect.BottomRight :=
if PtInRect(DockCenterRect,	Point(ClientWidth div 5 * 4,
MousePos) then	ClientHeight div 5 * 4);
begin	
Result := alClient;	//Find out where the mouse cursor
DockRect :=	is, to decide where to draw dock
DockCenterRect;	preview.
end;	if PtInRect(DockLeftRect,
if Result = alNone then Exit;	MousePos) then
ii itesuit aii (one then Eart)	begin
//DockRect is in screen coordinates.	Result := alLeft;
DockRect.TopLeft :=	DockRect := DockLeftRect;
ClientToScreen(DockRect.TopLeft);	DockRect.Right := ClientWidth
DockRect.BottomRight :=	div 2;
ClientToScreen(DockRect.BottomRi	end
ght);	else
end;	if PtInRect(DockTopRect,
· · · · · · · · · · · · · · · · · · ·	MousePos) then
procedure	begin
TDockableForm.FormClose(Sender:	Result := alTop;
TObject;	DockRect := DockTopRect;
var Action: TCloseAction);	DockRect.Left := 0;
begin	DockRect.Right :=
//the action taken depends on how	ClientWidth;
the form is docked.	DockRect.Bottom :=
the form is docted.	

ManualDock can be safely called	
during a drag	if (HostDockSite is
//operation is we override	TConjoinDockHost) then
processing of CM_DOCKCLIENT.	begin
	//remove the form's caption from
if Message.DockSource.Control is	the conjoin dock host's caption list
TDockableForm then	
begin	TConjoinDockHost(HostDockSite).U
begin	pdateCaption(Self);
//Find out how to dock (Using a	puate Caption (Sen);
TAlign as the result of	//if we're the last visible form on a
ComputeDockingRect)	conjoined form, hide the form
•	if
Pt.x := Message.MousePos.x;	
Pt.y := Message.MousePos.y;	HostDockSite.VisibleDockClientCou
DockType :=	nt <= 1 then
ComputeDockingRect(ARect, Pt);	HostDockSite.Hide;
	end;
//if we are over a dockable form	
docked to a panel in the	//if docked to a panel, tell the panel
//main window, manually dock the	to hide itself. If there are other
dragged form to the panel with	//visible dock clients on the panel, it
//the correct orientation.	ShowDockPanel won't allow it to
if (HostDockSite is TPanel) then	//be hidden
begin	if (HostDockSite is TPanel) then
Message.DockSource.Control.Manu	MainForm.ShowDockPanel(HostDo
Message.DockSource.Control.Manu	MainForm.ShowDockPanel(HostDockSite as TPanel, False, nil):
alDock(HostDockSite, nil,	MainForm.ShowDockPanel(HostDockSite as TPanel, False, nil);
alDock(HostDockSite, nil, DockType);	ckSite as TPanel, False, nil);
alDock(HostDockSite, nil, DockType); Exit;	ckSite as TPanel, False, nil);  Action := caHide;
alDock(HostDockSite, nil, DockType);	ckSite as TPanel, False, nil);
alDock(HostDockSite, nil, DockType); Exit; end;	ckSite as TPanel, False, nil);  Action := caHide; end;
alDock(HostDockSite, nil, DockType); Exit; end; //alClient => Create a	ckSite as TPanel, False, nil);  Action := caHide; end;  procedure
alDock(HostDockSite, nil, DockType); Exit; end;  //alClient => Create a TabDockHost and manually dock	ckSite as TPanel, False, nil);  Action := caHide; end;  procedure TDockableForm.CMDockClient(var
alDock(HostDockSite, nil, DockType); Exit; end;  //alClient => Create a TabDockHost and manually dock both forms to the PageControl	ckSite as TPanel, False, nil);  Action := caHide; end;  procedure TDockableForm.CMDockClient(var Message: TCMDockClient);
alDock(HostDockSite, nil, DockType); Exit; end;  //alClient => Create a TabDockHost and manually dock both forms to the PageControl //owned by the TabDockHost.	ckSite as TPanel, False, nil);  Action := caHide; end;  procedure TDockableForm.CMDockClient(var Message: TCMDockClient); var
alDock(HostDockSite, nil, DockType); Exit; end;  //alClient => Create a TabDockHost and manually dock both forms to the PageControl //owned by the TabDockHost. if DockType = alClient then	ckSite as TPanel, False, nil);  Action := caHide; end;  procedure TDockableForm.CMDockClient(var Message: TCMDockClient); var ARect: TRect;
alDock(HostDockSite, nil, DockType); Exit; end;  //alClient => Create a TabDockHost and manually dock both forms to the PageControl //owned by the TabDockHost. if DockType = alClient then begin	ckSite as TPanel, False, nil);  Action := caHide; end;  procedure TDockableForm.CMDockClient(var Message: TCMDockClient); var ARect: TRect; DockType: TAlign;
alDock(HostDockSite, nil, DockType); Exit; end;  //alClient => Create a TabDockHost and manually dock both forms to the PageControl //owned by the TabDockHost. if DockType = alClient then begin Host :=	ckSite as TPanel, False, nil);  Action := caHide; end;  procedure TDockableForm.CMDockClient(var Message: TCMDockClient); var ARect: TRect; DockType: TAlign; Host: TForm;
alDock(HostDockSite, nil, DockType); Exit; end;  //alClient => Create a TabDockHost and manually dock both forms to the PageControl //owned by the TabDockHost. if DockType = alClient then begin Host := TTabDockHost.Create(Application);	ckSite as TPanel, False, nil);  Action := caHide; end;  procedure TDockableForm.CMDockClient(var Message: TCMDockClient); var ARect: TRect; DockType: TAlign; Host: TForm; Pt: TPoint;
alDock(HostDockSite, nil, DockType); Exit; end;  //alClient => Create a TabDockHost and manually dock both forms to the PageControl //owned by the TabDockHost. if DockType = alClient then begin Host := TTabDockHost.Create(Application); Host.BoundsRect :=	ckSite as TPanel, False, nil);  Action := caHide; end;  procedure TDockableForm.CMDockClient(var Message: TCMDockClient); var ARect: TRect; DockType: TAlign; Host: TForm; Pt: TPoint; begin
alDock(HostDockSite, nil, DockType); Exit; end;  //alClient => Create a TabDockHost and manually dock both forms to the PageControl //owned by the TabDockHost. if DockType = alClient then begin Host := TTabDockHost.Create(Application);	ckSite as TPanel, False, nil);  Action := caHide; end;  procedure TDockableForm.CMDockClient(var Message: TCMDockClient); var ARect: TRect; DockType: TAlign; Host: TForm; Pt: TPoint; begin //Overriding this message allows
alDock(HostDockSite, nil, DockType); Exit; end;  //alClient => Create a TabDockHost and manually dock both forms to the PageControl //owned by the TabDockHost. if DockType = alClient then begin Host := TTabDockHost.Create(Application); Host.BoundsRect := Self.BoundsRect;	ckSite as TPanel, False, nil);  Action := caHide; end;  procedure TDockableForm.CMDockClient(var Message: TCMDockClient); var ARect: TRect; DockType: TAlign; Host: TForm; Pt: TPoint; begin //Overriding this message allows the dock form to create host forms
alDock(HostDockSite, nil, DockType); Exit; end;  //alClient => Create a TabDockHost and manually dock both forms to the PageControl //owned by the TabDockHost. if DockType = alClient then begin Host := TTabDockHost.Create(Application); Host.BoundsRect := Self.BoundsRect;  Self.ManualDock(TTabDockHost(H	ckSite as TPanel, False, nil);  Action := caHide; end;  procedure TDockableForm.CMDockClient(var Message: TCMDockClient); var ARect: TRect; DockType: TAlign; Host: TForm; Pt: TPoint; begin //Overriding this message allows
alDock(HostDockSite, nil, DockType); Exit; end;  //alClient => Create a TabDockHost and manually dock both forms to the PageControl //owned by the TabDockHost. if DockType = alClient then begin Host := TTabDockHost.Create(Application); Host.BoundsRect := Self.BoundsRect;	ckSite as TPanel, False, nil);  Action := caHide; end;  procedure TDockableForm.CMDockClient(var Message: TCMDockClient); var ARect: TRect; DockType: TAlign; Host: TForm; Pt: TPoint; begin //Overriding this message allows the dock form to create host forms
alDock(HostDockSite, nil, DockType); Exit; end;  //alClient => Create a TabDockHost and manually dock both forms to the PageControl //owned by the TabDockHost. if DockType = alClient then begin Host := TTabDockHost.Create(Application); Host.BoundsRect := Self.BoundsRect;  Self.ManualDock(TTabDockHost(H	ckSite as TPanel, False, nil);  Action := caHide; end;  procedure TDockableForm.CMDockClient(var Message: TCMDockClient); var ARect: TRect; DockType: TAlign; Host: TForm; Pt: TPoint; begin //Overriding this message allows the dock form to create host forms //depending on the mouse position
alDock(HostDockSite, nil, DockType); Exit; end;  //alClient => Create a TabDockHost and manually dock both forms to the PageControl //owned by the TabDockHost. if DockType = alClient then begin Host := TTabDockHost.Create(Application); Host.BoundsRect := Self.BoundsRect;  Self.ManualDock(TTabDockHost(H	ckSite as TPanel, False, nil);  Action := caHide; end;  procedure TDockableForm.CMDockClient(var Message: TCMDockClient); var ARect: TRect; DockType: TAlign; Host: TForm; Pt: TPoint; begin //Overriding this message allows the dock form to create host forms //depending on the mouse position when docking occurs. If we don't
alDock(HostDockSite, nil, DockType); Exit; end;  //alClient => Create a TabDockHost and manually dock both forms to the PageControl //owned by the TabDockHost. if DockType = alClient then begin Host := TTabDockHost.Create(Application); Host.BoundsRect := Self.BoundsRect;  Self.ManualDock(TTabDockHost(H ost).PageControl1, nil, alClient);	ckSite as TPanel, False, nil);  Action := caHide; end;  procedure TDockableForm.CMDockClient(var Message: TCMDockClient); var ARect: TRect; DockType: TAlign; Host: TForm; Pt: TPoint; begin //Overriding this message allows the dock form to create host forms //depending on the mouse position when docking occurs. If we don't override
alDock(HostDockSite, nil, DockType); Exit; end;  //alClient => Create a TabDockHost and manually dock both forms to the PageControl //owned by the TabDockHost. if DockType = alClient then begin Host := TTabDockHost.Create(Application); Host.BoundsRect := Self.BoundsRect;  Self.ManualDock(TTabDockHost(H ost).PageControl1, nil, alClient);  Message.DockSource.Control.Manu	ckSite as TPanel, False, nil);  Action := caHide; end;  procedure TDockableForm.CMDockClient(var Message: TCMDockClient); var ARect: TRect; DockType: TAlign; Host: TForm; Pt: TPoint; begin //Overriding this message allows the dock form to create host forms //depending on the mouse position when docking occurs. If we don't override //this message, the form will use
alDock(HostDockSite, nil, DockType); Exit; end;  //alClient => Create a TabDockHost and manually dock both forms to the PageControl //owned by the TabDockHost. if DockType = alClient then begin Host := TTabDockHost.Create(Application); Host.BoundsRect := Self.BoundsRect;  Self.ManualDock(TTabDockHost(H ost).PageControl1, nil, alClient);  Message.DockSource.Control.Manu alDock(TTabDockHost(Host).PageC	ckSite as TPanel, False, nil);  Action := caHide; end;  procedure TDockableForm.CMDockClient(var Message: TCMDockClient); var ARect: TRect; DockType: TAlign; Host: TForm; Pt: TPoint; begin //Overriding this message allows the dock form to create host forms //depending on the mouse position when docking occurs. If we don't override //this message, the form will use

waa	and
uses	end
Windows, Messages, SysUtils,	//if DockType <> alClient, create
Classes, Graphics, Controls, Forms,	the ConjoinDockHost and manually
Dialogs,	dock both
ComCtrls;	//forms to it. Be sure to make
	dockable forms non-dockable when
type	hosted by
TTabDockHost = class(TForm)	// ConjoinDockForm, since it is
PageControl1: TPageControl;	using the VCL default
<> procedure	DockManager.
FormClose(Sender: TObject; var	else begin
Action: TCloseAction);	Host :=
<> procedure	TConjoinDockHost.Create(Applicati
PageControl1UnDock(Sender:	on);
TObject; Client: TControl;	Host.BoundsRect :=
NewTarget: TWinControl; var	Self.BoundsRect;
,	Self.ManualDock(Host, nil,
Allow: Boolean);	alNone);
<pre>&gt; procedure PageControllCotSiteInfo(Sondon)</pre>	
PageControl1GetSiteInfo(Sender:	Self.DockSite := False;
TObject;	M P IG G ( )
DockClient: TControl; var	Message.DockSource.Control.Manu
InfluenceRect: TRect; MousePos:	alDock(Host, nil, DockType);
TPoint;	
var CanDock: Boolean);	TDockableForm(Message.DockSour
<> procedure	ce.Control).DockSite := False;
PageControl1DockOver(Sender:	Host.Visible := True;
TObject;	end;
Source: TDragDockObject; X,	end;
Y: Integer; State: TDragState;	end;
var Accept: Boolean);	, and the second
private	procedure
P	TDockableForm.FormShow(Sender:
{ Private declarations }	TObject);
public	begin
-	if HostDockSite is
{ Public declarations }	TConjoinDockHost then
end;	1 ConjoinDockflost then
	TConicinDeal-Heat/Heat/Heat/Heat/Heat/Heat/Heat/Heat/
var	TConjoinDockHost(HostDockSite).U
TabDockHost: TTabDockHost;	pdateCaption(nil);
	end;
implementation	
	end.
{\$R *.dfm}	\\\\\\\
uses uDockForm;	
	unit uTabHost;
procedure	Ź
TTabDockHost.FormClose(Sender:	interface
`	1110111100
TObject;	

procedure	
TTabDockHost.PageC	Control1DockO
ver(Sender: TObject;	
Source: TDragDock(	Object; X, Y:
Integer; State: TDrag	State;
var Accept: Boolean)	);
begin	
Accept := Source.Co	ntrol is
TDockableForm;	
end;	
end.	
<i> </i>	

<> Object \$94	
الهلفالنصي لبرناهج	
الهندانتظي ببرتاهم	
(Docking)	
إعداد	
علاء الدين اللباد	
واتف ۷۱۳۵۷۵۶۹۰	
<> Object MainForm:	
TMainForm	
Left = -4	
Top = -4	
Width = 1032	
Height = 746	
Caption = 'Docking Demo'	
Color = clWindow	
ParentFont = True	
Menu = MainMenu1	
OldCreateOrder = False	
OnCreate = FormCreate	
PixelsPerInch = 96	
TextHeight = 13	
<> Object VSplitter:	
TSplitter	
Left = 0	
Top = 52	
Width = 4	
Height = 636	

var Action: TCloseAction);
var
ARect: TRect;
begin
if PageControl1.DockClientCount =
1 then
begin
with PageControl1.DockClients[0]
do
begin
ARect.TopLeft :=
ClientToScreen(Point(0, 0));
ARect.BottomRight :=
ClientToScreen(Point(UndockWidth
, UndockHeight));
ManualFloat(ARect);
end;
Action := caFree;
end else
Action := caHide;
end;
procedure
TTabDockHost.PageControl1UnDoc
k(Sender: TObject;
Client: TControl; NewTarget:
TWinControl; var Allow: Boolean);
begin
//only 2 dock clients means the host
must be destroyed and
//the remaining window undocked
to its old position and size.
if (PageControl1.DockClientCount
= 2) and (NewTarget <> Self) then
PostMessage(Self.Handle,
WM_CLOSE, 0, 0);
end;
procedure
TTabDockHost.PageControl1GetSit
eInfo(Sender: TObject;
DockClient: TControl; var
InfluenceRect: TRect; MousePos:
TPoint;
var CanDock: Boolean);
begin
CanDock := DockClient is
TDockableForm;
end;

Caption = 'ToolBar1'
Constraints.MaxWidth = 280
DragKind = dkDock
DragMode = dmAutomatic
Flat = True
ShowCaptions = True
TabOrder = 0
Transparent = True
Wrapable = False
<> Object ToolButton13: TToolButton
Left = 0
Top = 0
Action = ExitAction
end
<> Object ToolButton16:
TToolButton
Left = 59
Top = 0
Width = 14
Caption = 'ToolButton16'
ImageIndex = 7
Style = tbsSeparator
end
<> Object btnToolBar1:
TToolButton
Left = 73
Top = 0
Action = ViewToolBar1
Style = tbsCheck
end
<> Object btnToolBar2:
TToolButton
Left = 132
Top = 0
Action = ViewToolBar2
Style = tbsCheck
end
end
<> Object ToolBar2:
TToolBar
Left = 9
Top = 25
1 op = 25 Width = 390
Height = 23
AutoSize = True
ButtonHeight = 21
ButtonWidth = 43

Visible = False
end
<> Object HSplitter:
TSplitter
Left = 0
Top = 688
Width = 1024
Height = 4
Cursor = crVSplit
Align = alBottom
Visible = False
end
<> Object CoolBar1:
TCoolBar
Left = 0
Top = 0
Width = 1024
Height = 52
AutoSize = True
BandMaximize = bmDblClick
Bands = <
item
Break = False
Control = ToolBar1
ImageIndex = -1
MinHeight = 23
Width = 1020
end
item
Control = ToolBar2
ImageIndex = -1
MinHeight = 23
Width = 1020
end>
Color = clMenu
DockSite = True
ParentColor = False
OnDockOver =
CoolBar1DockOver
<> Object ToolBar1:
TToolBar
Left = 9
Top = 0
Width = 280
Height = 23
AutoSize = True
ButtonHeight = 21
ButtonWidth = 59

Top = 0
Action = ViewTealWindow
end
end
end
<> Object LeftDockPanel:
TPanel
Left = 4
Top = 52
Width = 0
Height = 636
Align = alLeft
DockSite = True
TabOrder = 1
OnDockDrop =
LeftDockPanelDockDrop
OnDockOver =
LeftDockPanelDockOver
OnGetSiteInfo =
LeftDockPanelGetSiteInfo
OnUnDock =
LeftDockPanelUnDock
end
<> Object BottomDockPanel:
TPanel
Left = 0
Top = 688
Width = 1024
Height = 0
Align = alBottom
DockSite = True
TabOrder = 2
OnDockDrop =
LeftDockPanelDockDrop
OnDockOver =
BottomDockPanelDockOver
OnGetSiteInfo =
<b>LeftDockPanelGetSiteInfo</b>
OnUnDock =
LeftDockPanelUnDock
end
<> Object ActionList1:
TActionList
Left = 136
Top = 80
<> Object ViewToolBar1:
TAction Cotogory - Wiew Tool Boys!
Category = 'ViewToolBars' Caption = 'ToolBar &1'
antion = 'Loolkar X-l'

Caption = 'ToolBar2'
Constraints.MaxWidth = 390
DragKind = dkDock
DragMode = dmAutomatic
Flat = True
ShowCaptions = True
TabOrder = 1
Transparent = True
Wrapable = False
<> Object ToolButton1:
TToolButton
Left = 0
Top = 0
Action = ViewWhiteWindow
end
<> Object ToolButton2:
TToolButton
Left = 43
Top = 0
Action = ViewBlueWindow
end
<> Object ToolButton3:
TToolButton
Left = 86
Top = 0
Action = ViewGreenWindow
end
<> Object ToolButton5:
TToolButton
Left = 129
Top = 0
Action = ViewLimeWindow
end
<> Object ToolButton6:
TToolButton
Left = 172
Top = 0
Action = ViewPurpleWindow
end
<> Object ToolButton7:
TToolButton
Left = 215
Top = 0
Action = ViewRedWindow
end
<> Object ToolButton4:
TToolButton
Left = 258
LCII = 430

ViewWhiteWindowExecute
end
<> Object
ViewTealWindow: TAction
Tag = 4
Category = 'ViewWindows'
Caption = '&Teal'
OnExecute =
ViewWhiteWindowExecute
end
<> Object
ViewPurpleWindow: TAction
Tag = 5
Category = 'ViewWindows'
Caption = '&Purple'
OnExecute =
ViewWhiteWindowExecute
end
<> Object
ViewLimeWindow: TAction
Tag = 6
Category = 'ViewWindows'
Caption = '&Lime'
OnExecute =
ViewWhiteWindowExecute
end
end
<> Object MainMenu1:
TMainMenu
Left = 176
Top = 80
<> Object File2:
TMenuItem
Caption = '&File'
<> Object Exit2:
TMenuItem
Action = ExitAction
end
end
<> Object View2:
TMenuItem
Caption = '&View'
<> Object ToolBar11:
TMenuItem
Action = ViewToolBar1
end
<> Object ToolBar21:
TMenuItem
Action = ViewToolBar2

Checked = True  ImageIndex = 1  OnExecute = ViewToolBar1Execute  end <> Object ViewToolBar2: TAction  Category = 'ViewToolBars'  Caption = 'ToolBar & 2'
OnExecute = ViewToolBar1Execute end <> Object ViewToolBar2: TAction Category = 'ViewToolBars'
ViewToolBar1Execute end <> Object ViewToolBar2: TAction Category = 'ViewToolBars'
end <> Object ViewToolBar2: TAction Category = 'ViewToolBars'
<> Object ViewToolBar2: TAction Category = 'ViewToolBars'
TAction  Category = 'ViewToolBars'
TAction  Category = 'ViewToolBars'
Category = 'ViewToolBars'
Labilon = Toolbal &2
Checked = True
ImageIndex = 2
OnExecute =
ViewToolBar2Execute
end
<> Object
ViewWhiteWindow: TAction
Category = 'ViewWindows'
Caption = '&White'
Hint = 'View white window'
OnExecute =
ViewWhiteWindowExecute
end
<> Object ExitAction:
TAction
Caption = 'E&xit'
OnExecute = ExitActionExecute
end
<> Object
ViewBlueWindow: TAction
Tag = 1
Category = 'ViewWindows'
Caption = '&Blue'
OnExecute =
ViewWhiteWindowExecute
end
<> Object
ViewGreenWindow: TAction
Tag = 2
Category = 'ViewWindows'
Caption = '&Green'
OnExecute =
ViewWhiteWindowExecute
end
<> Object
ViewRedWindow: TAction
Tag = 3
Tag = 3

Font.Height = -10
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = False
OnClose = FormClose
PixelsPerInch = 96
TextHeight = 13
<> Object PageControl1:
TPageControl
Left = 0
Top = 0
Width = 453
Height = 243
Align = alClient
DockSite = True
TabOrder = 0
OnDockOver =
PageControl1DockOver
OnGetSiteInfo =
PageControl1GetSiteInfo
OnUnDock =
PageControl1UnDock
end
End
\\\\\\\\
<> Object ConjoinDockHost:
TConjoinDockHost
Left = 152
Top = 118
Width = 606
Height = 337
BorderStyle = bsSizeToolWin
Caption = 'ConjoinDockHost'
Color = clBtnFace
UseDockManager = True
DockSite = True
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -10
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = False
OnClose = FormClose
OnDockDrop = FormDockDrop
OnDockOver = FormDockOver
OnGetSiteInfo = FormGetSiteInfo
OnUnDock = FormUnDock

end
<> Object N2: TMenuItem
Caption = '-'
end
<> Object White1:
TMenuItem
Action = ViewWhiteWindow
end
<> Object Blue1:
TMenuItem
Action = ViewBlueWindow
end
<> Object Green1:
TMenuItem
Action = ViewGreenWindow
end
<> Object Lime1:
TMenuItem
Action = ViewLimeWindow
end
<> Object Purple1:
TMenuItem
Action = ViewPurpleWindow
end
<> Object Red1:
TMenuItem
Action = ViewRedWindow
_
end
<> Object Teal1:
TMenuItem
Action = ViewTealWindow
end
end
end
End
\\\
<> Object TabDockHost:
TTabDockHost
Left = 184
Top = 119
Width = 461
Width = 401 Height = 277
Caption = 'TabDockHost'
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText

Caption = 'Docking Demo'
Color = clWindow
ParentFont = True
Menu = MainMenu1
OldCreateOrder = False
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> Object VSplitter:
TSplitter
Left = 0
Top = 52
Width = 4
Height = 388
Visible = False
end
<> Object HSplitter:
TSplitter
Left = 0
Top = 440
Width = 675
Height = 4
Cursor = crVSplit
Align = alBottom
Visible = False
end
<> Object CoolBar1:
TCoolBar
Left = 0
Top = 0
Width = 675
Height = 52
AutoSize = True
BandMaximize = bmDblClick
Bands = <
item
Break = False
Control = ToolBar1
ImageIndex = -1
MinHeight = 23
Width = 671
end
item
Control = ToolBar2
ImageIndex = -1
MinHeight = 23
Width = 671
end>

PixelsPerInch = 96
TextHeight = 13 End
Enu
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
<> Object DockableForm:
TDockableForm
Left = 138
Top = 115
Width = 397
Height = 241
Caption = 'DockableForm'
Color = clBtnFace
DockSite = True
DragKind = dkDock
DragMode = dmAutomatic
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -10
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = False
OnClose = FormClose
OnDockOver = FormDockOver
OnShow = FormShow
PixelsPerInch = 96
TextHeight = 13
<> Object Memo1: TMemo
Left = 0
Top = 0
Width = 389
Height = 207
Align = alClient
Lines.Strings = (
'Memo1')
TabOrder = 0
end
End
111111111111111111111111111111111111111
<> Object MainForm:
TMainForm
Left = 181
Top = 152
Width = 683
Height = 498

Top = 0
Action = ViewToolBar2
Style = tbsCheck
end
end
<> Object ToolBar2:
TToolBar
Left = 9
Top = 25
Width = 390
Height = 23
AutoSize = True
ButtonHeight = 21
ButtonWidth = 43
Caption = 'ToolBar2'
Constraints.MaxWidth = 390
DragKind = dkDock
DragMode = dmAutomatic
Flat = True
ShowCaptions = True
TabOrder = 1
Transparent = True
Wrapable = False
<> Object ToolButton1:
TToolButton
Left = 0
Top = 0
Action = ViewWhiteWindow
end
<> Object ToolButton2:
TToolButton
Left = 43
Top = 0
Action = ViewBlueWindow
end
<> Object ToolButton3:
TToolButton
Left = 86
Top = 0
Action = ViewGreenWindow
end
<> Object ToolButton5:
TToolButton
Left = 129
Top = 0
Action = ViewLimeWindow
end
<> Object ToolButton6:

Color = clMenu
DockSite = True
ParentColor = False
OnDockOver =
CoolBar1DockOver
<> Object ToolBar1:
TToolBar
Left = 9
Top = 0
Width = 280
Height = 23
AutoSize = True
ButtonHeight = 21 ButtonWidth = 59
Caption = 'ToolBar1'
Constraints.MaxWidth = 280
DragKind = dkDock
DragMode = dmAutomatic
Flat = True
ShowCaptions = True
TabOrder = 0
Transparent = True
Wrapable = False
<> Object ToolButton13:
TToolButton
TToolButton Left = 0
Left = 0
Left = 0 Top = 0
Left = 0 Top = 0 Action = ExitAction end
Left = 0 Top = 0 Action = ExitAction
Left = 0 Top = 0 Action = ExitAction end <> Object ToolButton16: TToolButton
Left = 0 Top = 0 Action = ExitAction end <> Object ToolButton16: TToolButton Left = 59
Left = 0 Top = 0 Action = ExitAction end <> Object ToolButton16: TToolButton Left = 59 Top = 0
Left = 0 Top = 0 Action = ExitAction end <> Object ToolButton16: TToolButton Left = 59 Top = 0 Width = 14
Left = 0 Top = 0 Action = ExitAction end <> Object ToolButton16: TToolButton Left = 59 Top = 0 Width = 14 Caption = 'ToolButton16'
Left = 0 Top = 0 Action = ExitAction end <> Object ToolButton16: TToolButton Left = 59 Top = 0 Width = 14 Caption = 'ToolButton16' ImageIndex = 7
Left = 0 Top = 0 Action = ExitAction end <> Object ToolButton16: TToolButton Left = 59 Top = 0 Width = 14 Caption = 'ToolButton16' ImageIndex = 7 Style = tbsSeparator
Left = 0 Top = 0 Action = ExitAction end <> Object ToolButton16: TToolButton Left = 59 Top = 0 Width = 14 Caption = 'ToolButton16' ImageIndex = 7 Style = tbsSeparator end
Left = 0 Top = 0 Action = ExitAction end <> Object ToolButton16: TToolButton Left = 59 Top = 0 Width = 14 Caption = 'ToolButton16' ImageIndex = 7 Style = tbsSeparator end <> Object btnToolBar1:
Left = 0 Top = 0 Action = ExitAction end <> Object ToolButton16: TToolButton Left = 59 Top = 0 Width = 14 Caption = 'ToolButton16' ImageIndex = 7 Style = tbsSeparator end <> Object btnToolBar1: TToolButton
Left = 0 Top = 0 Action = ExitAction end <> Object ToolButton16: TToolButton Left = 59 Top = 0 Width = 14 Caption = 'ToolButton16' ImageIndex = 7 Style = tbsSeparator end <> Object btnToolBar1: TToolButton Left = 73
Left = 0 Top = 0 Action = ExitAction end <> Object ToolButton16: TToolButton Left = 59 Top = 0 Width = 14 Caption = 'ToolButton16' ImageIndex = 7 Style = tbsSeparator end <> Object btnToolBar1: TToolButton Left = 73 Top = 0
Left = 0 Top = 0 Action = ExitAction end <> Object ToolButton16: TToolButton Left = 59 Top = 0 Width = 14 Caption = 'ToolButton16' ImageIndex = 7 Style = tbsSeparator end <> Object btnToolBar1: TToolButton Left = 73 Top = 0 Action = ViewToolBar1
Left = 0 Top = 0 Action = ExitAction end <> Object ToolButton16: TToolButton Left = 59 Top = 0 Width = 14 Caption = 'ToolButton16' ImageIndex = 7 Style = tbsSeparator end <> Object btnToolBar1: TToolButton Left = 73 Top = 0 Action = ViewToolBar1 Style = tbsCheck
Left = 0 Top = 0 Action = ExitAction end <> Object ToolButton16: TToolButton Left = 59 Top = 0 Width = 14 Caption = 'ToolButton16' ImageIndex = 7 Style = tbsSeparator end <> Object btnToolBar1: TToolButton Left = 73 Top = 0 Action = ViewToolBar1 Style = tbsCheck end
Left = 0 Top = 0 Action = ExitAction end <> Object ToolButton16: TToolButton Left = 59 Top = 0 Width = 14 Caption = 'ToolButton16' ImageIndex = 7 Style = tbsSeparator end <> Object btnToolBar1: TToolButton Left = 73 Top = 0 Action = ViewToolBar1 Style = tbsCheck end <> Object btnToolBar2:
Left = 0 Top = 0 Action = ExitAction end <> Object ToolButton16: TToolButton Left = 59 Top = 0 Width = 14 Caption = 'ToolButton16' ImageIndex = 7 Style = tbsSeparator end <> Object btnToolBar1: TToolButton Left = 73 Top = 0 Action = ViewToolBar1 Style = tbsCheck end

OnDockOver =
BottomDockPanelDockOver
OnGetSiteInfo =
LeftDockPanelGetSiteInfo
OnUnDock =
LeftDockPanelUnDock
end
<> Object ActionList1:
TActionList
Left = 136
Top = 80
<> Object ViewToolBar1:
TAction
Category = 'ViewToolBars'
Caption = 'ToolBar &1'
Checked = True
ImageIndex = 1
OnExecute =
ViewToolBar1Execute
end
<> Object ViewToolBar2:
TAction
Category = 'ViewToolBars'
Caption = 'ToolBar &2'
Checked = True
ImageIndex = 2
OnExecute =
ViewToolBar2Execute
end
<> Object
ViewWhiteWindow: TAction
Category = 'ViewWindows'
Caption = '&White'
Hint = 'View white window'
OnExecute =
ViewWhiteWindowExecute
end
<> Object ExitAction:
TAction
Caption = 'E&xit'
OnExecute = ExitActionExecute
end
<> Object
ViewBlueWindow: TAction
Tag = 1
Category = 'ViewWindows'
Caption = '&Blue'
OnExecute =
ViewWhiteWindowExecute
. 10 ii ii mite ii muu ii Liactutt

TToolButton  Left = 172  Top = 0  Action = ViewPurpleWindow end  <> Object ToolButton7:  TToolButton  Left = 215  Top = 0  Action = ViewRedWindow end  <> Object ToolButton4:  TToolButton  Left = 258
Top = 0 Action = ViewPurpleWindow end <> Object ToolButton7: TToolButton Left = 215 Top = 0 Action = ViewRedWindow end <> Object ToolButton4: TToolButton Left = 258
Action = ViewPurpleWindow end <> Object ToolButton7: TToolButton Left = 215 Top = 0 Action = ViewRedWindow end <> Object ToolButton4: TToolButton Left = 258
end <> Object ToolButton7: TToolButton  Left = 215  Top = 0  Action = ViewRedWindow end <> Object ToolButton4: TToolButton  Left = 258
end <> Object ToolButton7: TToolButton  Left = 215  Top = 0  Action = ViewRedWindow end <> Object ToolButton4: TToolButton  Left = 258
TToolButton  Left = 215  Top = 0  Action = ViewRedWindow end  <> Object ToolButton4:  TToolButton  Left = 258
TToolButton  Left = 215  Top = 0  Action = ViewRedWindow end  <> Object ToolButton4:  TToolButton  Left = 258
Left = 215  Top = 0  Action = ViewRedWindow end <> Object ToolButton4: TToolButton Left = 258
Top = 0 Action = ViewRedWindow end <> Object ToolButton4: TToolButton Left = 258
Action = ViewRedWindow end <> Object ToolButton4: TToolButton Left = 258
end <> Object ToolButton4: TToolButton Left = 258
<> Object ToolButton4: TToolButton Left = 258
TToolButton Left = 258
Left = 258
Ton = 0
Top = 0
Action = ViewTealWindow
end
end
end
<> Object LeftDockPanel:
TPanel
Left = 4
Top = 52
Width = 0
Height = 388
Align = alLeft
DockSite = True
TabOrder = 1
OnDockDrop =
LeftDockPanelDockDrop
OnDockOver =
LeftDockPanelDockOver
OnGetSiteInfo =
LeftDockPanelGetSiteInfo
OnUnDock =
LeftDockPanelUnDock
end
<> Object BottomDockPanel:
TPanel
Left = 0
Top = 440
Width = 675
Height = 0
Align = alBottom
DockSite = True
TabOrder = 2
OnDockDrop =
LeftDockPanelDockDrop

<> Object Exit2:
TMenuItem
Action = ExitAction
end
end
<> Object View2:
TMenuItem
Caption = '&View'
<> Object ToolBar11:
TMenuItem
Action = ViewToolBar1
end
<> Object ToolBar21:
TMenuItem
Action = ViewToolBar2
end
<> Object N2: TMenuItem
Caption = '-'
end
<> Object White1:
TMenuItem
Action = ViewWhiteWindow
end
<> Object Blue1:
TMenuItem
Action = ViewBlueWindow
end
<> Object Green1: TMenuItem
Action = ViewGreenWindow
end
<> Object Lime1: TMenuItem
Action = ViewLimeWindow
end <> Object Purple1:
TMenuItem
Action = ViewPurpleWindow
end
<> Object Red1:
TMenuItem
Action = ViewRedWindow
end
<> Object Teal1:
TMenuItem
Action = ViewTealWindow
end
end
end

end
<> Object
ViewGreenWindow: TAction
Tag = 2
Category = 'ViewWindows'
Caption = '&Green'
OnExecute =
ViewWhiteWindowExecute
end
<> Object
ViewRedWindow: TAction
Tag = 3
Category = 'ViewWindows'
Caption = '&Red'
OnExecute =
ViewWhiteWindowExecute
end
<> Object
ViewTealWindow: TAction
Tag = 4
Category = 'ViewWindows'
Caption = '&Teal'
OnExecute =
ViewWhiteWindowExecute
end
<> Object
ViewPurpleWindow: TAction
Tag = 5
Category = 'ViewWindows'
Caption = '&Purple'
OnExecute =
ViewWhiteWindowExecute
end
<> Object
ViewLimeWindow: TAction
Tag = 6
Category = 'ViewWindows'
Caption = '&Lime'
OnExecute =
ViewWhiteWindowExecute
end
end
<> Object MainMenu1:
TMainMenu
Left = 176
Top = 80
<> Object File2:
TMenuItem
Caption = '&File'

Editor: TRichEdit;
StatusBar: TStatusBar;
StandardToolBar: TToolBar;
OpenButton: TToolButton;
SaveButton: TToolButton;
PrintButton: TToolButton;
ToolButton5: TToolButton;
UndoButton: TToolButton;
CutButton: TToolButton;
CopyButton: TToolButton;
PasteButton: TToolButton;
ToolButton10: TToolButton;
FontName: TComboBox;
FontSize: TEdit;
ToolButton11: TToolButton;
UpDown1: TUpDown;
BoldButton: TToolButton;
ItalicButton: TToolButton;
UnderlineButton: TToolButton;
ToolButton16: TToolButton;
LeftAlign: TToolButton;
CenterAlign: TToolButton;
RightAlign: TToolButton;
ToolButton20: TToolButton;
BulletsButton: TToolButton;
ToolbarImages: TImageList;
ActionList1: TActionList;
FileNewCmd: TAction;
FileOpenCmd: TAction;
FileSaveCmd: TAction;
FilePrintCmd: TAction;
FileExitCmd: TAction;
ToolButton1: TToolButton;
ToolButton2: TToolButton;
Bevel1: TBevel;
LanguageMenu: TMenuItem;
LanguageEnglish: TMenuItem;
LanguageGerman: TMenuItem;
EditCutCmd: TAction;
EditCopyCmd: TAction;
EditPasteCmd: TAction;
EditUndoCmd: TAction;
EditFontCmd: TAction;
FileSaveAsCmd: TAction;
LanguageFrench: TMenuItem;
<> procedure
SelectionChange(Sender: TObject);

Procedure§90		
الملف التنفيذي لبرنامج		
(, 0 ) (, 0 )		
(RichEdit)		
إعداد		
علاء الدين اللباد		
واتة. ۷۱۳۵۷۵۶۹۰ واته کارپور		
unit remain;		
interface		
uses		
SysUtils, Windows, Messages,		
Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Buttons,		
ExtCtrls, Menus, ComCtrls,		
ClipBrd,		
ToolWin, ActnList, ImgList;		
type		
TMainForm = class(TForm)		
MainMenu: TMainMenu;		
FileNewItem: TMenuItem;		
FileOpenItem: TMenuItem; FileSaveItem: TMenuItem;		
FileSaveAsItem: TMenuItem;		
FilePrintItem: TMenuItem;		
FileExitItem: TMenuItem;		
EditUndoItem: TMenuItem;		
EditCutItem: TMenuItem;		
EditCopyItem: TMenuItem;		
EditPasteItem: TMenuItem;		
HelpAboutItem: TMenuItem; OpenDialog: TOpenDialog;		
SaveDialog: TSaveDialog;		
PrintDialog: TPrintDialog;		
Ruler: TPanel;		
FontDialog1: TFontDialog;		
FirstInd: TLabel;		
LeftInd: TLabel; RulerLine: TBevel;		
RightInd: TLabel;		
N5: TMenuItem;		
miEditFont: TMenuItem;		

D. H. A. D. A. C. C. L. C. L.	
BulletsButtonClick(Sender:	<> procedure
TObject);	FormCreate(Sender: TObject);
<> procedure	<> procedure
FormCloseQuery(Sender: TObject;	ShowHint(Sender: TObject);
var CanClose: Boolean);	<> procedure
<> procedure	FileNew(Sender: TObject);
RulerItemMouseDown(Sender:	<> procedure
TObject; Button: TMouseButton;	FileOpen(Sender: TObject);
Shift: TShiftState; X, Y:	<> procedure
Integer);	FileSave(Sender: TObject);
<> procedure	<> procedure
RulerItemMouseMove(Sender:	FileSaveAs(Sender: TObject);
TObject; Shift: TShiftState; X,	<> procedure
Y: Integer);	FilePrint(Sender: TObject);
<> procedure	<> procedure
FirstIndMouseUp(Sender: TObject;	FileExit(Sender: TObject);
Button: TMouseButton;	<> procedure
Shift: TShiftState; X, Y:	EditUndo(Sender: TObject);
Integer);	<> procedure
<> procedure	EditCut(Sender: TObject);
LeftIndMouseUp(Sender: TObject;	
Button: TMouseButton;	<> procedure EditConv(Condon TObicat)
Shift: TShiftState; X, Y:	EditCopy(Sender: TObject);
	<> procedure
Integer);	EditPaste(Sender: TObject);
<> procedure	<> procedure
RightIndMouseUp(Sender: TObject; Button: TMouseButton;	HelpAbout(Sender: TObject);
	<> procedure
Shift: TShiftState; X, Y:	SelectFont(Sender: TObject);
Integer);	<> procedure
<> procedure	RulerResize(Sender: TObject);
FormShow(Sender: TObject);	<> procedure
<> procedure	FormResize(Sender: TObject);
RichEditChange(Sender: TObject);	<> procedure
<> procedure	FormPaint(Sender: TObject);
SwitchLanguage(Sender: TObject);	<> procedure
<> procedure	BoldButtonClick(Sender: TObject);
ActionList2Update(Action:	<> procedure
TBasicAction;	ItalicButtonClick(Sender: TObject);
var Handled: Boolean);	<> procedure
private	FontSizeChange(Sender: TObject);
private FFileName: string;	FontSizeChange(Sender: TObject); <> procedure
private FFileName: string; FUpdating: Boolean;	FontSizeChange(Sender: TObject); <> procedure AlignButtonClick(Sender: TObject);
private FFileName: string; FUpdating: Boolean; FDragOfs: Integer;	FontSizeChange(Sender: TObject); <> procedure AlignButtonClick(Sender: TObject); <> procedure
private FFileName: string; FUpdating: Boolean;	FontSizeChange(Sender: TObject); <> procedure AlignButtonClick(Sender: TObject);
private FFileName: string; FUpdating: Boolean; FDragOfs: Integer;	FontSizeChange(Sender: TObject); <> procedure AlignButtonClick(Sender: TObject); <> procedure
private FFileName: string; FUpdating: Boolean; FDragOfs: Integer; FDragging: Boolean;	FontSizeChange(Sender: TObject);  <> procedure AlignButtonClick(Sender: TObject);  <> procedure FontNameChange(Sender:
private FFileName: string; FUpdating: Boolean; FDragOfs: Integer; FDragging: Boolean; function CurrText:	FontSizeChange(Sender: TObject);  <> procedure AlignButtonClick(Sender: TObject);  <> procedure FontNameChange(Sender: TObject);
private FFileName: string; FUpdating: Boolean; FDragOfs: Integer; FDragging: Boolean; function CurrText: TTextAttributes;	FontSizeChange(Sender: TObject);  <> procedure AlignButtonClick(Sender: TObject);  <> procedure FontNameChange(Sender: TObject);  <> procedure
private FFileName: string; FUpdating: Boolean; FDragOfs: Integer; FDragging: Boolean; function CurrText: TTextAttributes; <> procedure	FontSizeChange(Sender: TObject);  <> procedure AlignButtonClick(Sender: TObject);  <> procedure FontNameChange(Sender: TObject);  <> procedure UnderlineButtonClick(Sender:

LANC CEDMAN.	C-AESL-NI (
LANG_GERMAN;	SetFileName(const FileName:
	String);
{\$R *.dfm}	<> procedure
	CheckFileSave;
<> procedure	<> procedure SetupRuler;
TMainForm.SelectionChange(Sende	<> procedure SetEditRect;
r: TObject);	<> procedure
begin	UpdateCursorPos;
with Editor.Paragraph do	<> procedure
try	WMDropFiles(var Msg:
FUpdating := True;	TWMDropFiles); message
FirstInd.Left :=	WM DROPFILES;
Trunc(FirstIndent*RulerAdj)-	<> procedure
4+GutterWid;	PerformFileOpen(const AFileName:
LeftInd.Left :=	string);
	<> procedure
Trunc((LeftIndent+FirstIndent)*Rul	SetModified(Value: Boolean);
erAdj)-4+GutterWid;	end;
RightInd.Left :=	ciiu,
Ruler.ClientWidth-6-	
Trunc((RightIndent+GutterWid)*R	var
ulerAdj);	MainForm: TMainForm;
BoldButton.Down := fsBold in	
Editor.SelAttributes.Style;	implementation
ItalicButton.Down := fsItalic in	
Editor.SelAttributes.Style;	uses REAbout, RichEdit, ShellAPI,
UnderlineButton.Down :=	ReInit;
fsUnderline in	
Editor.SelAttributes.Style;	resourcestring
BulletsButton.Down :=	sSaveChanges = 'Save changes to
Boolean(Numbering);	%s?';
FontSize.Text :=	sOverWrite = 'OK to overwrite
IntToStr(Editor.SelAttributes.Size);	%s';
FontName.Text :=	sUntitled = 'Untitled';
Editor.SelAttributes.Name;	sModified = 'Modified';
case Ord(Alignment) of	sColRowInfo = 'Line: %3d Col:
0: LeftAlign.Down := True;	%3d';
1: RightAlign.Down := True;	/03u ,
2: CenterAlign.Down := True;	agnet
	Const
end;	RulerAdj = 4/3;
UpdateCursorPos;	GutterWid = 6;
finally	
FUpdating := False;	ENGLISH =
end;	(SUBLANG_ENGLISH_US shl 10)
end;	or LANG_ENGLISH;
	FRENCH =
function TMainForm.CurrText:	(SUBLANG_FRENCH shl 10) or
TTextAttributes;	LANG_FRENCH;
begin	GERMAN =
if Editor.SelLength > 0 then Result	(SUBLANG GERMAN shl 10) or
- a - /	

mbYesNoCancel, 0);	:= Editor.SelAttributes
case SaveResp of	<pre>else Result := Editor.DefAttributes;</pre>
idYes: FileSave(Self);	end;
idNo: {Nothing};	
idCancel: Abort;	function EnumFontsProc(var
end;	LogFont: TLogFont; var
end;	TextMetric: TTextMetric;
	FontType: Integer; Data: Pointer):
<> procedure	Integer; stdcall;
TMainForm.SetupRuler;	begin
var	
I: Integer;	TStrings(Data).Add(LogFont.lfFace
S: String;	Name);
begin	<b>Result</b> := 1;
SetLength(S, 201);	end;
I := 1;	
while I < 200 do	<> procedure
begin	TMainForm.GetFontNames;
S[I] := #9;	var
S[I+1] := ' ';	DC: HDC;
Inc(I, 2);	begin
end;	DC := GetDC(0);
Ruler.Caption := S;	EnumFonts(DC, nil,
end;	@EnumFontsProc,
enu;	Pointer(FontName.Items));
Z Name and have	ReleaseDC(0, DC);
<> procedure TMainForm.SetEditRect;	FontName.Sorted := True;
,	end;
var	Citu,
R: TRect;	<> procedure
begin	TMainForm.SetFileName(const
with Editor do	FileName: String);
begin	begin
R := Rect(GutterWid, 0,	FFileName := FileName;
ClientWidth-GutterWid,	Caption := Format('%s - %s',
ClientHeight);	- · · · · · · · · · · · · · · · · · · ·
SendMessage(Handle,	[ExtractFileName(FileName),
<pre>EM_SETRECT, 0, Longint(@R));</pre>	Application.Title]);
end;	end;
end;	
	<> procedure
{ Event Handlers }	TMainForm.CheckFileSave;
	var
<> procedure	SaveResp: Integer;
TMainForm.FormCreate(Sender:	begin
TObject);	if not Editor. Modified then Exit;
begin	SaveResp :=
Application.OnHint := ShowHint;	MessageDlg(Format(sSaveChanges,
OpenDialog.InitialDir :=	[FFileName]),
	mtConfirmation,

SetFileName(sUntitled);	ExtractFilePath(ParamStr(0));
Editor.Lines.Clear;	SaveDialog.InitialDir :=
Editor.Modified := False;	OpenDialog.InitialDir;
· · ·	SetFileName(sUntitled);
SetModified(False);	GetFontNames;
end;	,
	SetupRuler;
<> procedure	SelectionChange(Self);
TMainForm.PerformFileOpen(const AFileName: string);	CurrText.Name :=
begin	DefFontData.Name;
begin	CurrText.Size := -
Editor Lines LoadEromFile(AFileNa	
Editor.Lines.LoadFromFile(AFileNa	MulDiv(DefFontData.Height, 72, Screen.PixelsPerInch);
me);	Screen.rixeisrerinch);
SetFileName(AFileName);	Language Fuelish Taga-
Editor.SetFocus;	LanguageEnglish.Tag := ENGLISH;
Editor.Modified := False;	,
SetModified(False);	LanguageFrench.Tag := FRENCH;
end;	LanguageGerman.Tag :=
	GERMAN;
<> procedure	case SysLocale.DefaultLCID of
TMainForm.FileOpen(Sender:	ENGLISH:
TObject);	SwitchLanguage(LanguageEnglish);
begin	FRENCH:
CheckFileSave;	SwitchLanguage(LanguageFrench); GERMAN:
if OpenDialog.Execute then	
begin	SwitchLanguage(LanguageGerman)
D	end;
PerformFileOpen(OpenDialog.FileN	end;
ame);	enu;
Editor.ReadOnly := ofReadOnly	/ > nwo.aoduwo
in OpenDialog.Options;	<> procedure TMainForm.ShowHint(Sender:
end;	TObject);
end;	begin
Z Spangadama	if Length(Application.Hint) > 0
<> procedure TMoinForm FiloSoyo(Sondom	then
TMainForm.FileSave(Sender:	begin
TObject);	StatusBar.SimplePanel := True;
begin  if FFiloNome – sUntitled then	StatusBar.SimpleTaner:= True; StatusBar.SimpleText :=
if FFileName = sUntitled then FileSeve As(Sender)	Application.Hint;
FileSaveAs(Sender)	end
else	else StatusBar.SimplePanel :=
begin	False;
Editor Lines SaveTaEtle (EEtleName)	end;
Editor.Lines.SaveToFile(FFileName)	cnu,
, Editor Modified :- Felse:	<> procedure
Editor.Modified := False;	TMainForm.FileNew(Sender:
SetModified(False);	TObject);
end;	** *
end;	begin

1	
end;	
	<> procedure
<> procedure	TMainForm.FileSaveAs(Sender:
TMainForm.EditCut(Sender:	TObject);
TObject);	begin
begin	if SaveDialog.Execute then
Editor.CutToClipboard;	begin
end;	if
	FileExists(SaveDialog.FileName)
<> procedure	then
TMainForm.EditCopy(Sender:	if
TObject);	MessageDlg(Format(sOverWrite,
begin	[SaveDialog.FileName]),
Editor.CopyToClipboard;	mtConfirmation,
end;	mbYesNoCancel, 0) <> idYes then
	Exit;
<> procedure	,
TMainForm.EditPaste(Sender:	Editor.Lines.SaveToFile(SaveDialog.
TObject);	FileName);
begin	
Editor.PasteFromClipboard;	SetFileName(SaveDialog.FileName);
end;	Editor. Modified := False;
enu;	SetModified(False);
	end;
<> procedure	end;
TMainForm.HelpAbout(Sender:	enu,
TObject);	Z Name and man
begin	<> procedure
with TAboutBox.Create(Self) do	TMainForm.FilePrint(Sender:
try	TObject);
ShowModal;	begin
finally	if PrintDialog.Execute then
Free;	Editor.Print(FFileName);
end;	end;
end;	
	<> procedure
<> procedure	TMainForm.FileExit(Sender:
TMainForm.SelectFont(Sender:	TObject);
TObject);	begin
begin	Close;
	end;
FontDialog1.Font.Assign(Editor.Sel	
Attributes);	<> procedure
if FontDialog1.Execute then	TMainForm.EditUndo(Sender:
in a companied supercontention	TObject);
CurrText.Assign(FontDialog1.Font);	begin
SelectionChange(Self);	with Editor do
Editor.SetFocus;	if HandleAllocated then
,	SendMessage(Handle, EM UNDO,
end;	0, 0);
	0, 0),

	<> procedure
<> procedure	TMainForm.RulerResize(Sender:
TMainForm.FontSizeChange(Sende	TObject);
r: TObject);	begin
begin	RulerLine.Width :=
if FUpdating then Exit;	Ruler.ClientWidth -
CurrText.Size :=	(RulerLine.Left*2);
	end;
StrToInt(FontSize.Text);	enu,
end;	< > nwoooduwo
	<pre>&lt;&gt; procedure TMainForm.FormResize(Sender:</pre>
<> procedure	`
TMainForm.AlignButtonClick(Send	TObject);
er: TObject);	begin Safe did Dagge
begin	SetEditRect;
if FUpdating then Exit;	SelectionChange(Sender);
Editor.Paragraph.Alignment :=	end;
TAlignment(TControl(Sender).Tag);	
end;	<> procedure
	TMainForm.FormPaint(Sender:
<> procedure	TObject);
TMainForm.FontNameChange(Send	begin
er: TObject);	SetEditRect;
begin	end;
if FUpdating then Exit;	
CurrText.Name :=	<> procedure
FontName.Items[FontName.ItemInd	TMainForm.BoldButtonClick(Sende
ex];	r: TObject);
end;	begin
	if FUpdating then Exit;
<> procedure	if BoldButton.Down then
TMainForm.UnderlineButtonClick(	CurrText.Style := CurrText.Style
Sender: TObject);	+ [fsBold]
begin	else
if FUpdating then Exit;	CurrText.Style := CurrText.Style
if UnderlineButton.Down then	- [fsBold];
CurrText.Style := CurrText.Style	end;
+ [fsUnderline]	
else	<> procedure
CurrText.Style := CurrText.Style	TMainForm.ItalicButtonClick(Send
- [fsUnderline];	er: TObject);
end;	begin
	if FUpdating then Exit;
<> procedure	if ItalicButton.Down then
TMainForm.BulletsButtonClick(Sen	CurrText.Style := CurrText.Style
der: TObject);	+ [fsItalic]
begin	else
if FUpdating then Exit;	CurrText.Style := CurrText.Style
Editor.Paragraph.Numbering :=	- [fsItalic];
TNumberingStyle(BulletsButton.Do	end;
	/

GutterWid) / RulerAdj);	wn);
LeftIndMouseUp(Sender, Button,	end;
Shift, X, Y);	cnu,
end;	< > nmaaaduma
enu,	<pre>&lt;&gt; procedure TMoinForm FormCloseQuery(Sand)</pre>
	TMainForm.FormCloseQuery(Send
<> procedure	er: TObject; var CanClose:
TMainForm.LeftIndMouseUp(Send	Boolean);
er: TObject; Button:	begin
TMouseButton;	try
Shift: TShiftState; X, Y: Integer);	CheckFileSave;
begin	except
FDragging := False;	CanClose := False;
Editor.Paragraph.LeftIndent :=	end;
Trunc((LeftInd.Left+FDragOfs-	end;
GutterWid) / RulerAdj)-	
Editor.Paragraph.FirstIndent;	{ Ruler Indent Dragging }
SelectionChange(Sender);	
end;	<> procedure
	TMainForm.RulerItemMouseDown(
<> procedure	Sender: TObject; Button:
TMainForm.RightIndMouseUp(Sen	TMouseButton;
der: TObject; Button:	Shift: TShiftState; X, Y: Integer);
TMouseButton;	begin
Shift: TShiftState; X, Y: Integer);	FDragOfs :=
begin	(TLabel(Sender).Width div 2);
FDragging := False;	TLabel(Sender).Left :=
Editor.Paragraph.RightIndent :=	TLabel(Sender).Left+X-FDragOfs;
Trunc((Ruler.ClientWidth-	FDragging := True;
RightInd.Left+FDragOfs-2)/	end;
RulerAdj)-2*GutterWid;	
SelectionChange(Sender);	<> procedure
end;	TMainForm.RulerItemMouseMove(
	Sender: TObject; Shift: TShiftState;
<> procedure	X, Y: Integer);
TMainForm.UpdateCursorPos;	begin
var	if FDragging then
CharPos: TPoint;	TLabel(Sender).Left :=
begin	TLabel(Sender).Left+X-FDragOfs
CharPos.Y :=	end;
SendMessage(Editor.Handle,	*****
EM_EXLINEFROMCHAR, 0,	<> procedure
Editor.SelStart);	TMainForm.FirstIndMouseUp(Send
CharPos.X := (Editor.SelStart -	er: TObject; Button:
SendMessage(Editor.Handle,	TMouseButton;
EM LINEINDEX, CharPos.Y, 0));	Shift: TShiftState; X, Y: Integer);
Inc(CharPos.Y);	begin
Inc(CharPos.X);	FDragging := False;
StatusBar.Panels[0].Text :=	
Format(sColRowInfo, [CharPos.Y,	Editor.Paragraph.FirstIndent :=
roi mausconcowinio, [Charros. 1,	Trunc((FirstInd.Left+FDragOfs-

if Value then
StatusBar.Panels[1].Text :=
sModified
else StatusBar.Panels[1].Text := '';
end;
<> procedure
TMainForm.SwitchLanguage(Sende
r: TObject);
var
Name: String;
Size : Integer;
begin
if
LoadNewResourceModule(TCompo
nent(Sender).Tag) <> 0 then
begin
Name := FontName.Text;
Size := StrToInt(FontSize.Text);
ReinitializeForms;
LanguageEnglish.Checked :=
LanguageEnglish = Sender;
LanguageFrench.Checked :=
LanguageFrench = Sender;
LanguageGerman.Checked :=
LanguageGerman = Sender;
CurrText.Name := Name;
CurrText.Size := Size;
SelectionChange(Self);
FontName.SelLength := 0;
SetupRuler;
if Visible then Editor.SetFocus;
end;
end;
<> procedure
TMainForm.ActionList2Update(Acti
on: TBasicAction;
var Handled: Boolean);
begin
{ Update the status of the edit
commands }
EditCutCmd.Enabled :=
Editor.SelLength > 0;
EditCopyCmd.Enabled :=
EditCutCmd.Enabled;
if Editor.HandleAllocated then

CharPos.X]);
end;
<> procedure
TMainForm.FormShow(Sender:
TObject);
begin
UpdateCursorPos;
DragAcceptFiles(Handle, True);
RichEditChange(nil);
Editor.SetFocus;
{ Check if we should load a file
from the command line }
if (ParamCount > 0) and
FileExists(ParamStr(1)) then
PerformFileOpen(ParamStr(1));
end;
,
<> procedure
TMainForm.WMDropFiles(var
Msg: TWMDropFiles);
var
CFileName: array[0MAX PATH]
of Char;
begin
if DragQueryFile(Msg.Drop, 0,
CFileName, MAX_PATH) > 0 then
begin CheckFileSave;
,
PerformFileOpen(CFileName);
Msg.Result := 0;
end;
finally
DragFinish(Msg.Drop);
end;
end;
_
<> procedure
TMainForm.RichEditChange(Sende
r: TObject);
begin
SetModified(Editor.Modified);
end;
<> procedure
TMainForm.SetModified(Value:
Boolean);
begin

<> procedure
TAboutBox.FormCreate(Sender:
TObject);
var
MS: TMemoryStatus;
begin
GlobalMemoryStatus(MS);
PhysMem.Caption :=
FormatFloat('#,###" KB"',
MS.dwTotalPhys / 1024);
FreeRes.Caption := Format('%d
%%', [MS.dwMemoryLoad]);
end;
end.
\\\\\\\

0=
<> Objects87
الهلف النصي لبرناهج
(RichEdit)
إعداد
علاء الدين اللباد
واتف ۷۱۳۵۷۵۶۹۰
<> Object MainForm:
TMainForm
Left = 173
Top = 156
Width = 625
Height = 209
ActiveControl = Editor
Caption = 'Rich Edit Control
Demo'
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'Default'
Font.Style = []
Icon.Data = {
FF}
Menu = MainMenu
OldCreateOrder = True

begin
EditUndoCmd.Enabled :=
Editor.Perform(EM CANUNDO, 0,
0) <> 0;
EditPasteCmd.Enabled :=
Editor.Perform(EM CANPASTE, 0,
0) > 0;
end;
end;
enu,
,
end.
\\\\\\\\\\\
unit reabout;
interface
uses Windows, Classes, Graphics,
Forms, Controls, StdCtrls,
Buttons, ExtCtrls, SysUtils;
Duttons, Externs, Systems,
two
type TAboutBox = class(TForm)
` ` `
OKButton: TButton;
ProgramIcon: TImage;
Label1: TLabel;
Bevel1: TBevel;
Label2: TLabel;
Label3: TLabel;
PhysMem: TLabel;
Label4: TLabel;
FreeRes: TLabel;
<> procedure
FormCreate(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
AboutBox: TAboutBox;
implementation
{\$R *.dfm}
,

I 6: 2
Left = 2
Top = 12
Width = 10
Height = 11
AutoSize = False
Caption = 'é'
DragCursor = crArrow
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'Wingdings'
Font.Style = []
ParentFont = False
OnMouseDown =
RulerItemMouseDown
OnMouseMove =
RulerItemMouseMove
OnMouseUp = LeftIndMouseUp
end
<> Object RulerLine:
TBevel
Left = 4
Top = 12
Width = 579
Height = 2
Shape = bsTopLine
end
<> Object RightInd: TLabel
Left = 575
Top = 14
Width = 9
Height = 12
Caption = 'ñ'
DragCursor = crArrow
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'Wingdings'
Font.Style = []
ParentFont = False
OnMouseDown =
RulerItemMouseDown
OnMouseMove =
RulerItemMouseMove
OnMouseUp =
RightIndMouseUp
•

Position = poDefaultSizeOnly
OnCloseQuery = FormCloseQuery
OnCreate = FormCreate
OnPaint = FormPaint
OnResize = FormResize
OnShow = FormShow
PixelsPerInch = 96
TextHeight = 13
<> Object Ruler: TPanel
Left = 0
Top = 32
Width = 617
Height = 26
Align = alTop
Alignment = taLeftJustify
BevelOuter = bvNone
Caption = 'asdf'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'Arial'
Font.Style = [] ParentFont = False
TabOrder = 0
OnResize = RulerResize
<> Object FirstInd: TLabel
Left = 2
Top = 2
Width = 10
Height = 9
AutoSize = False
Caption = 'ê'
DragCursor = crArrow
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'Wingdings'
Font.Style = []
ParentFont = False
OnMouseDown =
RulerItemMouseDown
OnMouseMove =
RulerItemMouseMove
OnMouseUp = FirstIndMouseUp
end
<> Object LeftInd: TLabel
S S Joer Deliting, I Dubel

Imagas - TaalhauImagas
Images = ToolbarImages
Indent = 4
ParentShowHint = False
ShowHint = True
TabOrder = 3
Wrapable = False
<> Object ToolButton1:
TToolButton
Left = 4
Top = 0
Action = FileNewCmd
end
<> Object OpenButton:
TToolButton
Left = 27
Top = 0
Action = FileOpenCmd
end
<> Object SaveButton:
TToolButton
Left = 50
Top = 0
Action = FileSaveCmd
end
<> Object PrintButton:
TToolButton
Left = 73
Top = 0
Action = FilePrintCmd
end
<> Object ToolButton5:
TToolButton
Left = 96
Top = 0
Width = 8
ImageIndex = 3
Style = tbsDivider
end
<> Object CutButton:
TToolButton
Left = 104
Top = 0
Action = EditCutCmd
end
<> Object CopyButton:
TToolButton
Left = 127
Top = 0

end
<> Object Bevel1: TBevel
Left = 0
Top = 0
Width = 617
Height = 2
Align = alTop
Shape = bsTopLine
end
end
<> Object Editor: TRichEdit
Left = 0
Top = 58
Width = 617
Height = 78
Align = alClient
ScrollBars = ssBoth
TabOrder = 1
OnChange = RichEditChange
OnSelectionChange =
SelectionChange
end
<> Object StatusBar:
TStatusBar
Left = 0
Top = 136
Width = 617
Height = 19
Panels = <
item
Width = 120
end
item
Alignment = taCenter Width = 60
end
item
Width = 50
end>
end
<> Object StandardToolBar:
TToolBar
Left = 0
Top = 0
Width = 617
AutoSize = True
BorderWidth = 2
Flat = True

Width = 26
Height = 22
Hint = 'Font Size Select font size'
TabOrder = 1
Text = '0'
OnChange = FontSizeChange
end
<> Object UpDown1:
TUpDown
Left = 411
Top = 0
Width = 15
Height = 22
Associate = FontSize
TabOrder = 2
end
<> Object ToolButton2:
TToolButton
Left = 426
Top = 0
Width = 8
Caption = 'ToolButton2'
ImageIndex = 15
Style = tbsSeparator
end
<> Object BoldButton:
TToolButton
Left = 434
Top = 0
Hint = 'Bold'
ImageIndex = 8
Style = tbsCheck
OnClick = BoldButtonClick
end
<> Object ItalicButton:
TToolButton
Left = 457
Top = 0
Hint = 'Italic'
ImageIndex = 9
Style = tbsCheck
OnClick = ItalicButtonClick
end
<> Object UnderlineButton:
TToolButton
Left = 480
Top = 0
Hint = 'Underline'

Action = EditCopyCmd
end
<> Object PasteButton:
TToolButton
Left = 150
Top = 0
Action = EditPasteCmd
end
<> Object UndoButton:
TToolButton
Left = 173
Top = 0
Action = EditUndoCmd
end
<> Object ToolButton10:
TToolButton
Left = 196
Top = 0
Width = 8
ImageIndex = 7
Style = tbsSeparator
end
<> Object FontName:
TComboBox
Left = 204
Top = 0
Width = 173
Height = 21
Hint = 'Font Name Select font
name'
Ctl3D = False
DropDownCount = 10
ItemHeight = 13
ParentCtl3D = False
TabOrder = 0
OnChange = FontNameChange
end
<> Object ToolButton11:
TToolButton
Left = 377
Top = 0
Width = 8
ImageIndex = 8
Style = tbsSeparator
end
<> Object FontSize: TEdit
Left = 385
Top = 0

Width = 8
ImageIndex = 15
Style = tbsDivider
end
<> Object BulletsButton:
TToolButton
Left = 588
Top = 0
Hint = 'Bullets Enter bullet
mode'
ImageIndex = 14
Style = tbsCheck
OnClick = BulletsButtonClick
end
end
<> Object MainMenu:
TMainMenu
Images = ToolbarImages
Left = 120
Top = 68
<> Object FileMenu:
TMenuItem
Caption = '&File'
<> Object FileNewItem:
TMenuItem
Action = FileNewCmd
end
<> Object FileOpenItem:
TMenuItem
Action = FileOpenCmd
end
<> Object FileSaveItem:
TMenuItem
Action = FileSaveCmd
end
<> Object FileSaveAsItem:
TMenuItem
Action = FileSaveAsCmd
end
<> Object N1: TMenuItem
Caption = '-'
end
<> Object FilePrintItem:
TMenuItem
Action = FilePrintCmd
end
<> Object N4: TMenuItem
Caption = '-'

ImageIndex = 10
Style = tbsCheck
OnClick = UnderlineButtonClick
end
<> Object ToolButton16:
TToolButton
Left = 503
Top = 0
Width = 8
ImageIndex = 12
Style = tbsDivider
- V
end
<> Object LeftAlign:
TToolButton
Left = 511
Top = 0
Hint = 'Align Left'
Grouped = True
ImageIndex = 11
Style = tbsCheck
OnClick = AlignButtonClick
end
<> Object CenterAlign:
TToolButton
Tag = 2
Left = 534
Top = 0
Hint = 'Center'
Grouped = True
ImageIndex = 12
Style = tbsCheck
OnClick = AlignButtonClick
end
<> Object RightAlign:
TToolButton
Tag = 1
Left = 557
Top = 0
Hint = 'Align Right'
Grouped = True
ImageIndex = 13
Style = tbsCheck
OnClick = AlignButtonClick
end
<> Object ToolButton20:
TToolButton
Left = 580
Top = 0
Toh o

Caption = '&French'
Hint = 'Switch translation to
French'
OnClick = SwitchLanguage
end
<> Object
LanguageGerman: TMenuItem
Caption = '&German'
Hint = 'Switch translation to
German'
OnClick = SwitchLanguage
end
end
<> Object HelpMenu:
TMenuItem
Caption = '&Help'
<> Object HelpAboutItem:
TMenuItem
Caption = '&About'
Hint = 'Show program
information'
OnClick = HelpAbout
end
end
end
<> Object OpenDialog:
TOpenDialog
Filter = 'Rich Text Files
(*.RTF) *.RTF Text Files
(*.TXT) *.TXT'
Left = 56
Top = 104
end
<> Object SaveDialog:
TSaveDialog
Filter = 'Rich Text Files
(*.RTF) *.RTF Text Files
(*.TXT) *.TXT'
Left = 88
Top = 104
end
<> Object PrintDialog:
TPrintDialog
Left = 120
Top = 104
end
<> Object FontDialog1:
TFontDialog
Font.Charset =
runt.Charset =

end
<> Object FileExitItem:
TMenuItem
Action = FileExitCmd
end
end
<> Object EditMenu:
TMenuItem
Caption = '&Edit'
<> Object EditUndoItem:
TMenuItem
Action = EditUndoCmd
end
<> Object N2: TMenuItem
Caption = '-'
end
<> Object EditCutItem:
TMenuItem
Action = EditCutCmd
end
<> Object EditCopyItem:
TMenuItem
Action = EditCopyCmd
end
<> Object EditPasteItem:
TMenuItem
Action = EditPasteCmd
end
<> Object N5: TMenuItem
Caption = '-'
end
<> Object miEditFont:
TMenuItem
Caption = '&Font'
OnClick = SelectFont
end
end
<> Object LanguageMenu:
TMenuItem
Caption = '&Language'
<> Object
LanguageEnglish: TMenuItem
Caption = '&English'
Hint = 'Switch translation to
English'
OnClick = SwitchLanguage
end
<> Object
LanguageFrench: TMenuItem

Category = 'File' Caption = '&Print' Hint = 'Print current file' ImageIndex = 3 ShortCut = 16464 OnExecute = FilePrint end <> Object FileExitCmd: TAction Category = 'File' Caption = 'E&xit' Hint = 'Exit this application' ShortCut = 32856 OnExecute = FileExit end <> Object FileSaveAsCmd: TAction Category = 'File' Caption = 'Save &As' Hint = 'Save current file under a new name' OnExecute = FileSaveAs end end <> Object ActionList2: TActionList Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472 OnExecute = EditCut	TAction
Caption = '&Print' Hint = 'Print current file' ImageIndex = 3 ShortCut = 16464 OnExecute = FilePrint end <> Object FileExitCmd: TAction Category = 'File' Caption = 'E&xit' Hint = 'Exit this application' ShortCut = 32856 OnExecute = FileExit end <> Object FileSaveAsCmd: TAction Category = 'File' Caption = 'Save &As' Hint = 'Save current file under a new name' OnExecute = FileSaveAs end end <> Object ActionList2: TActionList Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
Hint = 'Print current file' ImageIndex = 3 ShortCut = 16464 OnExecute = FilePrint end <> Object FileExitCmd: TAction Category = 'File' Caption = 'E&xit' Hint = 'Exit this application' ShortCut = 32856 OnExecute = FileExit end <> Object FileSaveAsCmd: TAction Category = 'File' Caption = 'Save &As' Hint = 'Save current file under a new name' OnExecute = FileSaveAs end end <> Object ActionList2: TActionList Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
ImageIndex = 3 ShortCut = 16464 OnExecute = FilePrint end <> Object FileExitCmd: TAction Category = 'File' Caption = 'E&xit' Hint = 'Exit this application' ShortCut = 32856 OnExecute = FileExit end <> Object FileSaveAsCmd: TAction Category = 'File' Caption = 'Save &As' Hint = 'Save current file under a new name' OnExecute = FileSaveAs end end <> Object ActionList2: TActionList Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
ShortCut = 16464 OnExecute = FilePrint end <> Object FileExitCmd: TAction Category = 'File' Caption = 'E&xit' Hint = 'Exit this application' ShortCut = 32856 OnExecute = FileExit end <> Object FileSaveAsCmd: TAction Category = 'File' Caption = 'Save &As' Hint = 'Save current file under a new name' OnExecute = FileSaveAs end end <> Object ActionList2: TActionList Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
OnExecute = FilePrint end  <> Object FileExitCmd: TAction  Category = 'File' Caption = 'E&xit' Hint = 'Exit this application' ShortCut = 32856 OnExecute = FileExit end  <> Object FileSaveAsCmd: TAction  Category = 'File' Caption = 'Save &As' Hint = 'Save current file under a new name' OnExecute = FileSaveAs end end  <> Object ActionList2: TActionList Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68  <> Object EditUndoCmd: TAction  Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end  <> Object EditCutCmd: TAction  Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
end  <> Object FileExitCmd: TAction  Category = 'File' Caption = 'E&xit' Hint = 'Exit this application' ShortCut = 32856 OnExecute = FileExit end <> Object FileSaveAsCmd: TAction  Category = 'File' Caption = 'Save &As' Hint = 'Save current file under a new name' OnExecute = FileSaveAs end end <> Object ActionList2: TActionList Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
Category = 'File' Caption = 'E&xit' Hint = 'Exit this application' ShortCut = 32856 OnExecute = FileExit end <> Object FileSaveAsCmd: TAction Category = 'File' Caption = 'Save &As' Hint = 'Save current file under a new name' OnExecute = FileSaveAs end end <> Object ActionList2: TActionList Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
TAction Category = 'File' Caption = 'E&xit' Hint = 'Exit this application' ShortCut = 32856 OnExecute = FileExit end <> Object FileSaveAsCmd: TAction Category = 'File' Caption = 'Save &As' Hint = 'Save current file under a new name' OnExecute = FileSaveAs end end <> Object ActionList2: TActionList Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
Caption = 'E&xit'  Hint = 'Exit this application' ShortCut = 32856 OnExecute = FileExit end <> Object FileSaveAsCmd: TAction Category = 'File' Caption = 'Save &As' Hint = 'Save current file under a new name' OnExecute = FileSaveAs end end <> Object ActionList2: TActionList Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
Caption = 'E&xit'  Hint = 'Exit this application' ShortCut = 32856 OnExecute = FileExit end <> Object FileSaveAsCmd: TAction Category = 'File' Caption = 'Save &As' Hint = 'Save current file under a new name' OnExecute = FileSaveAs end end <> Object ActionList2: TActionList Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	Category = 'File'
Hint = 'Exit this application' ShortCut = 32856 OnExecute = FileExit end <	
ShortCut = 32856 OnExecute = FileExit end <	
OnExecute = FileExit end <> Object FileSaveAsCmd: TAction Category = 'File' Caption = 'Save &As' Hint = 'Save current file under a new name' OnExecute = FileSaveAs end end <> Object ActionList2: TActionList Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
end <> Object FileSaveAsCmd: TAction  Category = 'File' Caption = 'Save &As'  Hint = 'Save current file under a new name' OnExecute = FileSaveAs end end <> Object ActionList2: TActionList  Images = ToolbarImages OnUpdate = ActionList2Update Left = 56  Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
<pre>Category = 'File' Caption = 'Save &amp;As' Hint = 'Save current file under a new name' OnExecute = FileSaveAs end end &lt;&gt; Object ActionList2: TActionList Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 &lt;&gt; Object EditUndoCmd: TAction Category = 'Edit' Caption = '&amp;Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end &lt;&gt; Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&amp;t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472</pre>	
TAction Category = 'File' Caption = 'Save &As' Hint = 'Save current file under a new name' OnExecute = FileSaveAs end end <> Object ActionList2: TActionList Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	<pre>&lt;&gt; Ohiect FileSaveAsCmd:</pre>
Caption = 'Save &As'  Hint = 'Save current file under a new name'  OnExecute = FileSaveAs end end <> Object ActionList2: TActionList  Images = ToolbarImages OnUpdate = ActionList2Update  Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	TAction
Hint = 'Save current file under a new name'  OnExecute = FileSaveAs end end <> Object ActionList2: TActionList  Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	Category = 'File'
Hint = 'Save current file under a new name'  OnExecute = FileSaveAs end end <> Object ActionList2: TActionList  Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	Caption = 'Save &As'
OnExecute = FileSaveAs end end <	
end  <> Object ActionList2: TActionList  Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	new name'
end  <> Object ActionList2: TActionList  Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	OnExecute = FileSaveAs
<> Object ActionList2: TActionList  Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
<> Object ActionList2: TActionList  Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	end
TActionList  Images = ToolbarImages  OnUpdate = ActionList2Update  Left = 56  Top = 68  <> Object EditUndoCmd:  TAction  Category = 'Edit'  Caption = '&Undo'  Hint = 'Undo the last action'  ImageIndex = 4  ShortCut = 16474  OnExecute = EditUndo  end  <> Object EditCutCmd:  TAction  Category = 'Edit'  Caption = 'Cu&t'  Hint = 'Delete selected item'  ImageIndex = 5  ShortCut = 16472	
Images = ToolbarImages OnUpdate = ActionList2Update Left = 56 Top = 68 <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	•
OnUpdate = ActionList2Update  Left = 56  Top = 68  <> Object EditUndoCmd:  TAction  Category = 'Edit'  Caption = '&Undo'  Hint = 'Undo the last action'  ImageIndex = 4  ShortCut = 16474  OnExecute = EditUndo  end  <> Object EditCutCmd:  TAction  Category = 'Edit'  Caption = 'Cu&t'  Hint = 'Delete selected item'  ImageIndex = 5  ShortCut = 16472	
Left = 56 Top = 68  <> Object EditUndoCmd: TAction Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	<u> </u>
Top = 68  <> Object EditUndoCmd: TAction  Category = 'Edit'  Caption = '&Undo'  Hint = 'Undo the last action'  ImageIndex = 4  ShortCut = 16474  OnExecute = EditUndo  end  <> Object EditCutCmd: TAction  Category = 'Edit'  Caption = 'Cu&t'  Hint = 'Delete selected item'  ImageIndex = 5  ShortCut = 16472	
<pre>&lt;&gt; Object EditUndoCmd: TAction     Category = 'Edit'     Caption = '&amp;Undo'     Hint = 'Undo the last action'     ImageIndex = 4     ShortCut = 16474     OnExecute = EditUndo     end     &lt;&gt; Object EditCutCmd: TAction     Category = 'Edit'     Caption = 'Cu&amp;t'     Hint = 'Delete selected item'     ImageIndex = 5     ShortCut = 16472</pre>	
TAction  Category = 'Edit'  Caption = '&Undo'  Hint = 'Undo the last action'  ImageIndex = 4  ShortCut = 16474  OnExecute = EditUndo  end  <> Object EditCutCmd:  TAction  Category = 'Edit'  Caption = 'Cu&t'  Hint = 'Delete selected item'  ImageIndex = 5  ShortCut = 16472	
Category = 'Edit' Caption = '&Undo' Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
Caption = '&Undo'  Hint = 'Undo the last action'  ImageIndex = 4  ShortCut = 16474  OnExecute = EditUndo end  <> Object EditCutCmd:  TAction  Category = 'Edit'  Caption = 'Cu&t'  Hint = 'Delete selected item'  ImageIndex = 5  ShortCut = 16472	
Hint = 'Undo the last action' ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
ImageIndex = 4 ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
ShortCut = 16474 OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
OnExecute = EditUndo end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
end <> Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
<pre>&lt;&gt; Object EditCutCmd: TAction Category = 'Edit' Caption = 'Cu&amp;t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472</pre>	
TAction  Category = 'Edit'  Caption = 'Cu&t'  Hint = 'Delete selected item'  ImageIndex = 5  ShortCut = 16472	
Category = 'Edit' Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
Caption = 'Cu&t' Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
Hint = 'Delete selected item' ImageIndex = 5 ShortCut = 16472	
ImageIndex = 5 ShortCut = 16472	-
ShortCut = 16472	
	ImageIndex = 5
OnExecute = EditCut	ShortCut = 16472
<u> </u>	OnExecute = EditCut

DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
Left = 24
Top = 104
end
<> Object ToolbarImages:
TImageList
Left = 88
Top = 68
-
Bitmap = {
00000000000000
end
<> Object ActionList1:
TActionList
Images = ToolbarImages
Left = 24
Top = 68
<> Object FileNewCmd:
TAction
Category = 'File'
Caption = '&New'
Hint = 'Create a new file'
ImageIndex = 0
ShortCut = 16462
OnExecute = FileNew
end
<> Object FileOpenCmd:
TAction
Category = 'File'
Caption = '&Open'
Hint = 'Open an existing file'
ImageIndex = 1
ShortCut = 16463
OnExecute = FileOpen
end
<> Object FileSaveCmd:
TAction
Category = 'File'
Caption = '&Save'
Hint = 'Save current file'
ImageIndex = 2
ShortCut = 16467
OnExecute = FileSave
end
<> Object FilePrintCmd:

Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> Object ProgramIcon:
TImage
Left = 8
Top = 12
Width = 64
Height = 64
AutoSize = True
Picture.Data = {
BFBF}
Transparent = True
IsControl = True
end
<> Object Label1: TLabel
Left = 81
Top = 22
Width = 143
Height = 13
Caption = 'Delphi Rich Edit
Control Demo'
end
<> Object Bevel1: TBevel
Left = 81
Top = 134
Width = 260
Height = 2
Shape = bsTopLine
end
<> Object Label2: TLabel
Left = 81
Top = 45
Width = 183
Height = 13
Caption = 'Copyright © 1998
Borland International'
end
<> Object Label3: TLabel
Left = 81
Top = 145
Width = 187
Height = 13
Caption = 'Physical Memory
Available to Windows:'
end
<> Object PhysMem: TLabel
Object i hyspitem. I Label

end
<> Object EditCopyCmd:
TAction
Category = 'Edit'
Caption = '&Copy'
Hint = 'Copy selected item to
clipboard'
ImageIndex = 6
ShortCut = 16451
OnExecute = EditCopy
end
<> Object EditPasteCmd:
TAction
Category = 'Edit'
Caption = '&Paste'
Hint = 'Paste contents of
clipboard'
ImageIndex = 7
ShortCut = 16470
OnExecute = EditPaste
end
<> Object EditFontCmd:
TAction
Category = 'Edit'
Caption = '&Font'
Caption = '&Font' OnExecute = SelectFont
OnExecute = SelectFont end
OnExecute = SelectFont
OnExecute = SelectFont end
OnExecute = SelectFont end end
OnExecute = SelectFont end end
OnExecute = SelectFont end end End
OnExecute = SelectFont
OnExecute = SelectFont end end End
OnExecute = SelectFont
OnExecute = SelectFont end end End
OnExecute = SelectFont end end End
OnExecute = SelectFont end end End  \text{\tinit}\text{\text{\text{\text{\text{\text{\text{\text{\tex{\tex
OnExecute = SelectFont end end End  \[\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
OnExecute = SelectFont end end End  \times
OnExecute = SelectFont end end End  Left = 183 Top = 438 BorderStyle = bsDialog Caption = 'About RichEdit' ClientHeight = 243 ClientWidth = 357
OnExecute = SelectFont end end End  William SelectFont end End  Color = clBtnFace
OnExecute = SelectFont end end End  William Select AboutBox: TAboutBox Left = 183 Top = 438 BorderStyle = bsDialog Caption = 'About RichEdit' ClientHeight = 243 ClientWidth = 357 Color = clBtnFace Font.Charset =
OnExecute = SelectFont end end End  Williminiminiminiminiminiminiminiminimini
OnExecute = SelectFont end end End  William Select AboutBox: TAboutBox Left = 183 Top = 438 BorderStyle = bsDialog Caption = 'About RichEdit' ClientHeight = 243 ClientWidth = 357 Color = clBtnFace Font.Charset =
OnExecute = SelectFont end end End  Williminiminiminiminiminiminiminiminimini
OnExecute = SelectFont end end End  William SelectFont  Color = clBtnFace Font.Color = clBlack
OnExecute = SelectFont end end End  Left = 183 Top = 438 BorderStyle = bsDialog Caption = 'About RichEdit' ClientHeight = 243 ClientWidth = 357 Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11
OnExecute = SelectFont end end End  Left = 183 Top = 438 BorderStyle = bsDialog Caption = 'About RichEdit' ClientHeight = 243 ClientWidth = 357 Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif'

uses
SysUtils, Windows, Messages,
Classes, Graphics, Controls, Forms,
Dialogs,
Menus, StdCtrls, ComCtrls;
type
TEditForm = class(TForm)
MainMenu1: TMainMenu;
File1: TMenuItem;
New1: TMenuItem;
Open1: TMenuItem;
Save1: TMenuItem;
Saveas1: TMenuItem;
Print1: TMenuItem;
N2: TMenuItem;
Exit1: TMenuItem;
Edit1: TMenuItem;
Cut1: TMenuItem;
Copy1: TMenuItem;
Paste1: TMenuItem;
Delete1: TMenuItem;
N3: TMenuItem;
Selectall1: TMenuItem;
Character1: TMenuItem;
Left1: TMenuItem;
Right1: TMenuItem;
Center1: TMenuItem;
N4: TMenuItem;
Wordwrap1: TMenuItem;
N5: TMenuItem;
Font1: TMenuItem;
Editor: TRichEdit;
PopupMenu1: TPopupMenu;
Cut2: TMenuItem;
Copy2: TMenuItem;
Paste2: TMenuItem;
SaveFileDialog: TSaveDialog;
FontDialog1: TFontDialog;
Printersetup1: TMenuItem;
Close1: TMenuItem;
PrinterSetupDialog1:
TPrinterSetupDialog;
PrintDialog1: TPrintDialog;
<>procedure
Exit1Click(Sender: TObject);
<>procedure
New1Click(Sender: TObject);

Left = 275
Top = 145
Width = 6
Height = 13
Caption = '0'
end
<> Object Label4: TLabel
Left = 81
<b>Top = 167</b>
Width = 73
Height = 13
Caption = 'Memory in Use:'
end
<> Object FreeRes: TLabel
Left = 276
Top = 167
Width = 6
Height = 13
Caption = '0'
end
<> Object OKButton:
TButton
Left = 269
Top = 208
Width = 75
Height = 25
Cancel = True
Caption = 'OK'
Default = True
ModalResult = 2
TabOrder = 0
end
End

ProcedureS54	
الملف التنفيذي لبرنامج	
(TextEdit)	
عداد	
علاء الدين اللباد	
هاتف ۷۹۲۵۷۵۳۷۱	
unit MDIEdit;	
interface	

EditForm: TEditForm;	<
	AlignC
const	<
DefaultFileName = 'Untitled';	Wordy
	<
implementation	Cut1C
	Conv.1
uses Clipbrd, Printers, MDIFrame;	Copy10
(AD 4 16 )	Paste1
{\$R *.dfm}	<
< >muses duns	Selecta
<>procedure	<
TEditForm.Exit1Click(Sender:	Delete1
TObject); begin	<
FrameForm.Exit1Click(Sender);	Edit1C
end;	<
enu,	Saveas
<>procedure	<
TEditForm.New1Click(Sender:	Save10
TObject);	<
begin	Font10
FrameForm.New1Click(Sender);	<
end;	Close1
- Cita;	<
<>procedure	FormC
TEditForm.Open1Click(Sender:	Action
TObject);	<
begin	FormC
FrameForm.Open1Click(Sender);	var Ca
end;	<
,	FormC
<>procedure	<
TEditForm.AlignClick(Sender:	Printer
TObject);	TObje
begin	 D41
Left1.Checked := False;	Print10
Right1.Checked := False;	<
Center1.Checked := False;	Open1
with Sender as TMenuItem do	privat
Checked := True;	{ Pri
with Editor.Paragraph do	Path
if Left1.Checked then	public
Alignment := taLeftJustify	{ Pu
else if Right1.Checked then	AFileN
Alignment := taRightJustify	end;
else if Center1.Checked then	l Enu;
Alignment := taCenter;	var
	var

<>procedure
AlignClick(Sender: TObject); <>procedure
Wordwrap1Click(Sender: TObject);
<>procedure
Cut1Click(Sender: TObject);
<>procedure
Copy1Click(Sender: TObject);
<>procedure
Paste1Click(Sender: TObject);
<>procedure
Selectall1Click(Sender: TObject);
<>procedure
Delete1Click(Sender: TObject);
<>procedure
Edit1Click(Sender: TObject);
<>procedure
=
Saveas1Click(Sender: TObject); <>procedure
Save1Click(Sender: TObject);
<>procedure
Font1Click(Sender: TObject);
<>procedure
Close1Click(Sender: TObject);
<>procedure
FormClose(Sender: TObject; var
Action: TCloseAction);
<>procedure
FormCloseQuery(Sender: TObject;
var CanClose: Boolean);
<>procedure
FormCreate(Sender: TObject);
<>procedure
Printersetup1Click(Sender:
TObject);
<>procedure
Print1Click(Sender: TObject);
<>procedure
Open1Click(Sender: TObject);
private
{ Private declarations }
PathName: string;
public
{ Public declarations }
<>procedure Open(const AFileName: string);
<u> </u>
end;
Van
var

TObject);	end;
begin	
Editor.ClearSelection;	<>procedure
end;	TEditForm.Wordwrap1Click(Sende
	r: TObject);
<>procedure	begin
TEditForm.Edit1Click(Sender:	with Editor do
TObject);	begin
var	WordWrap := not WordWrap; {
HasSelection: Boolean;	toggle word wrapping }
begin	if WordWrap then
Paste1.Enabled :=	ScrollBars := ssVertical
Clipboard.HasFormat(CF TEXT);	else
Paste2.Enabled := Paste1.Enabled;	ScrollBars := ssBoth;
HasSelection := Editor.SelLength >	WordWrap1.Checked :=
0;	WordWrap; { set menu item check }
Cut1.Enabled := HasSelection;	end;
Cut2.Enabled := HasSelection;	end;
Copy1.Enabled := HasSelection;	
Copy2.Enabled := HasSelection;	<>procedure
Delete1.Enabled := HasSelection;	TEditForm.Cut1Click(Sender:
end;	TObject);
Circi,	begin
<>procedure	Editor.CutToClipboard;
TEditForm.Open(const AFileName:	end;
string);	chu,
begin	<>procedure
PathName := AFileName;	TEditForm.Copy1Click(Sender:
Caption :=	TObject);
ExtractFileName(AFileName);	begin
with Editor do	Editor.CopyToClipboard;
begin	end;
Lines.LoadFromFile(PathName);	chu,
SelStart := 0;	<>procedure
· ·	TEditForm.Paste1Click(Sender:
Modified := False;	TObject);
end;	begin
end;	Editor.PasteFromClipboard;
	end;
<>procedure	enu,
TEditForm.Saveas1Click(Sender:	< >nuo anduno
TObject);	<>procedure
begin  SavaFilaDialog FilaNama	TEditForm.Selectall1Click(Sender: TObject);
SaveFileDialog.FileName :=	• /
PathName;	begin Editor Soloot Alle
if SaveFileDialog.Execute then	Editor.SelectAll;
begin	end;
PathName :=	
SaveFileDialog.FileName;	<>procedure
Caption :=	TEditForm.Delete1Click(Sender:

const	ExtractFileName(PathName);
SWarningText = 'Save changes to	Save1Click(Sender);
%s?';	end;
begin	end;
if Editor.Modified then	
begin	<>procedure
case	TEditForm.Save1Click(Sender:
MessageDlg(Format(SWarningText,	TObject);
[PathName]), mtConfirmation,	begin
[mbYes, mbNo, mbCancel], 0) of	if PathName = DefaultFileName
idYes: Save1Click(Self);	then
idCancel: CanClose := False;	SaveAs1Click(Sender)
end;	else
end;	begin
end;	
,	Editor.Lines.SaveToFile(PathName)
<>procedure	;
TEditForm.FormCreate(Sender:	Editor.Modified := False;
TObject);	end;
begin	end;
PathName := DefaultFileName;	,
end;	<>procedure
,	TEditForm.Font1Click(Sender:
<>procedure	TObject);
TEditForm.Printersetup1Click(Send	begin
er: TObject);	FontDialog1.Font := Editor.Font;
begin	if FontDialog1.Execute then
PrinterSetupDialog1.Execute;	
end;	Editor.SelAttributes.Assign(FontDia
,	log1.Font);
<>procedure	end;
TEditForm.Print1Click(Sender:	,
TObject);	<>procedure
begin	TEditForm.Close1Click(Sender:
if PrintDialog1.Execute then	TObject);
Editor.Print(PathName);	begin
end;	Close;
	end;
end.	,
	<>procedure
\\\\\\\	TEditForm.FormClose(Sender:
	TObject; var Action: TCloseAction);
	begin
unit MDIFrame;	Action := caFree;
unit Milli i anic,	end;
interface	7
IIICI IACC	<>procedure
11505	TEditForm.FormCloseQuery(Sende
uses	r: TObject; var CanClose: Boolean);

begin
Close;
end;
<>procedure
TFrameForm.New1Click(Sender:
TObject);
begin
TEditForm.Create(Self);
end;
<>procedure
TFrameForm.Tile1Click(Sender:
TObject);
begin
Tile;
end;
<>procedure
TFrameForm.Cascade1Click(Sender
: TObject);
begin
Cascade;
end;
<>procedure
TFrameForm.Arrangeicons1Click(S
ender: TObject);
begin
ArrangeIcons;
end;
<>procedure
TFrameForm.Open1Click(Sender:
TObject);
begin
if OpenFileDialog.Execute then
with TEditForm.Create(Self) do
Open(OpenFileDialog.FileName);
end;
end.
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

Classes, Graphics, Controls, Forms
Dialogs,
Menus;
type
TFrameForm = class(TForm)
MainMenu1: TMainMenu;
File1: TMenuItem;
New1: TMenuItem;
Open1: TMenuItem;
N1: TMenuItem;
Exit1: TMenuItem;
Window1: TMenuItem;
Tile1: TMenuItem;
Cascade1: TMenuItem;
Arrangeicons1: TMenuItem;
OpenFileDialog: TOpenDialog;
<>procedure
Exit1Click(Sender: TObject);
<>procedure
New1Click(Sender: TObject);
<>procedure
Tile1Click(Sender: TObject);
<>procedure
Cascade1Click(Sender: TObject);
<>procedure
Arrangeicons1Click(Sender:
TObject);
<>procedure
Open1Click(Sender: TObject);
(Private declarations)
{ Private declarations }
public { Public declarations }
end;
ciiu,
var
FrameForm: TFrameForm;
Trancroim, Irranicroim,
implementation
uses MDIEdit;
7
{\$R *.dfm}
,
<>procedure
TFrameForm.Exit1Click(Sender:
TObject);

Top = 248
<>Object File1: TMenuItem
Caption = '&File'
<>Object New1:
TMenuItem
Caption = '&New'
OnClick = New1Click
end
<>Object Open1:
TMenuItem
Caption = '&Open'
OnClick = Open1Click
end
<>Object Close1:
TMenuItem
Caption = '&Close'
OnClick = Close1Click
end
<>Object Save1:
TMenuItem
Caption = '&Save'
OnClick = Save1Click
end
<>Object Saveas1:
TMenuItem
Caption = 'Save &as'
OnClick = Saveas1Click
end
<>Object Print1:
TMenuItem
Caption = '&Print'
OnClick = Print1Click
end
<>Object Printersetup1:
TMenuItem
Caption = 'Printer setup'
OnClick = Printersetup1Click
end
<>Object N2: TMenuItem
Caption = '-'
end
<>Object Exit1:
TMenuItem
Caption = 'E&xit'
OnClick = Exit1Click
end
end end
<>Object Edit1: TMenuItem

<>Objects48
الملف لبرنامج
(TextEdit)
عامدا
علاء الدين اللباد
واتف ۷۱۳۵۷۵۶۹۰ واتف ۱۷۳۵۷۵۶۹۰
<>Object EditForm:
TEditForm
Left = 196
Top = 112
Width = 435
Height = 308
Caption = 'Untitled'
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET Font.Color = clWindowText
Font.Height = -11 Font.Name = 'MS Sans Serif'
Font.Style = []
FormStyle = fsMDIChild Menu = MainMenu1
OldCreateOrder = True
PopupMenu = PopupMenu1
Position = poDefault
Visible = True
OnClose = FormClose
OnCloseQuery = FormCloseQuery
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<>Object Editor: TRichEdit
Left = 0
Top = 0
Width = 427
Height = 254
Align = alClient
BorderStyle = bsNone
ScrollBars = ssVertical
TabOrder = 0
end
<>Object MainMenu1:
TMainMenu
Left = 16

TMenuItem
Caption = '&Right'
OnClick = AlignClick
end
<>Object Center1:
TMenuItem
Caption = '&Center'
OnClick = AlignClick
end
<>Object N4: TMenuItem
Caption = '-'
end
<>Object Wordwrap1:
TMenuItem
Caption = '&Word wrap'
Checked = True
OnClick = Wordwrap1Click
end
<>Object N5: TMenuItem
Caption = '-'
end
<>Object Font1:
TMenuItem
Caption = '&Font'
OnClick = Font1Click
end
end
end
<>Object PopupMenu1:
TPopupMenu
OnPopup = Edit1Click
Left = 96
Top = 64
<>Object Cut2: TMenuItem
Caption = 'Cu&t'
OnClick = Cut1Click
end
<>Object Copy2:
TMenuItem
Caption = '&Copy'
OnClick = Copy1Click
end
<>Object Paste2:
TMenuItem
Caption = '&Paste'
OnClick = Paste1Click
end
end
·

Caption = '&Edit'
GroupIndex = 1
OnClick = Edit1Click
<>Object Cut1:
TMenuItem
Caption = 'Cu&t'
ShortCut = 16472
OnClick = Cut1Click
end
<>Object Copy1:
TMenuItem
Caption = '&Copy'
ShortCut = 16451
OnClick = Copy1Click
end
<>Object Paste1:
TMenuItem
Caption = '&Paste'
<b>ShortCut = 16470</b>
OnClick = Paste1Click
end
<>Object Delete1:
TMenuItem
Caption = 'De&lete'
ShortCut = 16452
OnClick = Delete1Click
end
<>Object N3: TMenuItem
Caption = '-'
end
<>Object Selectall1:
TMenuItem
Caption = 'Select &all'
ShortCut = 16449
OnClick = Selectall1Click
end
end
<>Object Character1:
TMenuItem
Caption = '&Character'
GroupIndex = 1
<>Object Left1:
TMenuItem
Caption = '&Left'
Checked = True
OnClick = AlignClick
end
<>Object Right1:

Font.Name = 'MS Sans Serif'  Font.Style = []  FormStyle = fsMDIForm  Menu = MainMenu1  OldCreateOrder = True  Position = poDefault  WindowMenu = Window1  PixelsPerInch = 96  TextHeight = 13  <>Object MainMenu1:  TMainMenu  Left = 8  Top = 248  <>Object File1: TMenuItem  Caption = '&File'  <>Object New1:  TMenuItem  Caption = '&New'  OnClick = New1Click  end  <>Object Open1:  TMenuItem  Caption = '&Open'
Font.Style = [] FormStyle = fsMDIForm Menu = MainMenu1 OldCreateOrder = True Position = poDefault WindowMenu = Window1 PixelsPerInch = 96 TextHeight = 13 <>Object MainMenu1: TMainMenu Left = 8 Top = 248 <>Object File1: TMenuItem Caption = '&File' <>Object New1: TMenuItem Caption = '&New' OnClick = New1Click end <>Object Open1: TMenuItem
FormStyle = fsMDIForm Menu = MainMenu1 OldCreateOrder = True Position = poDefault WindowMenu = Window1 PixelsPerInch = 96 TextHeight = 13 <>Object MainMenu1: TMainMenu Left = 8 Top = 248 <>Object File1: TMenuItem Caption = '&File' <>Object New1: TMenuItem Caption = '&New' OnClick = New1Click end <>Object Open1: TMenuItem
Menu = MainMenu1 OldCreateOrder = True Position = poDefault WindowMenu = Window1 PixelsPerInch = 96 TextHeight = 13 <>Object MainMenu1: TMainMenu Left = 8 Top = 248 <>Object File1: TMenuItem Caption = '&File' <>Object New1: TMenuItem Caption = '&New' OnClick = New1Click end <>Object Open1: TMenuItem
OldCreateOrder = True Position = poDefault WindowMenu = Window1 PixelsPerInch = 96 TextHeight = 13 <>Object MainMenu1: TMainMenu Left = 8 Top = 248 <>Object File1: TMenuItem Caption = '&File' <>Object New1: TMenuItem Caption = '&New' OnClick = New1Click end <>Object Open1: TMenuItem
Position = poDefault WindowMenu = Window1 PixelsPerInch = 96 TextHeight = 13 <>Object MainMenu1: TMainMenu Left = 8 Top = 248 <>Object File1: TMenuItem Caption = '&File' <>Object New1: TMenuItem Caption = '&New' OnClick = New1Click end <>Object Open1: TMenuItem
WindowMenu = Window1 PixelsPerInch = 96 TextHeight = 13 <>Object MainMenu1: TMainMenu Left = 8 Top = 248 <>Object File1: TMenuItem Caption = '&File' <>Object New1: TMenuItem Caption = '&New' OnClick = New1Click end <>Object Open1: TMenuItem
PixelsPerInch = 96  TextHeight = 13  <>Object MainMenu1:  TMainMenu  Left = 8  Top = 248  <>Object File1: TMenuItem  Caption = '&File'  <>Object New1:  TMenuItem  Caption = '&New'  OnClick = New1Click  end  <>Object Open1:  TMenuItem
TextHeight = 13 <>Object MainMenu1: TMainMenu Left = 8 Top = 248 <>Object File1: TMenuItem Caption = '&File' <>Object New1: TMenuItem Caption = '&New' OnClick = New1Click end <>Object Open1: TMenuItem
<pre>&lt;&gt;Object MainMenu1: TMainMenu Left = 8 Top = 248 &lt;&gt;Object File1: TMenuItem Caption = '&amp;File' &lt;&gt;Object New1: TMenuItem Caption = '&amp;New' OnClick = New1Click end &lt;&gt;Object Open1: TMenuItem</pre>
TMainMenu  Left = 8  Top = 248  <>Object File1: TMenuItem  Caption = '&File'  <>Object New1:  TMenuItem  Caption = '&New'  OnClick = New1Click  end  <>Object Open1:  TMenuItem
Left = 8 Top = 248 <>Object File1: TMenuItem Caption = '&File' <>Object New1: TMenuItem Caption = '&New' OnClick = New1Click end <>Object Open1: TMenuItem
Top = 248  <>Object File1: TMenuItem Caption = '&File' <>Object New1: TMenuItem Caption = '&New' OnClick = New1Click end <>Object Open1: TMenuItem
<pre>&lt;&gt;Object File1: TMenuItem    Caption = '&amp;File'    &lt;&gt;Object New1: TMenuItem    Caption = '&amp;New'    OnClick = New1Click    end    &lt;&gt;Object Open1: TMenuItem</pre>
<pre>&lt;&gt;Object File1: TMenuItem    Caption = '&amp;File'    &lt;&gt;Object New1: TMenuItem    Caption = '&amp;New'    OnClick = New1Click    end    &lt;&gt;Object Open1: TMenuItem</pre>
Caption = '&File' <>Object New1: TMenuItem Caption = '&New' OnClick = New1Click end <>Object Open1: TMenuItem
<>Object New1: TMenuItem     Caption = '&New'     OnClick = New1Click     end     <>Object Open1: TMenuItem
TMenuItem Caption = '&New' OnClick = New1Click end <>Object Open1: TMenuItem
OnClick = New1Click end <>Object Open1: TMenuItem
OnClick = New1Click end <>Object Open1: TMenuItem
end <>Object Open1: TMenuItem
<>Object Open1: TMenuItem
TMenuItem
OnClick = Open1Click
end
<>Object N1: TMenuItem
Caption = '-'
end
<>Object Exit1:
TMenuItem
Caption = 'E&xit'
GroupIndex = 9
OnClick = Exit1Click
end
end
<>Object Window1:
TMenuItem
Caption = '&Window'
GroupIndex = 9
<>Object Tile1:
TMenuItem
Caption = '&Tile'
OnClick = Tile1Click
end
<>Object Cascade1:
TMenuItem
Caption = '&Cascade'

<>Object SaveFileDialog:
TSaveDialog
Filter =
'Rich text files (*.rtf) *.rtf Plain
text files (*.txt) *.txt All' +
' files *.*'
Options = [ofHideReadOnly,
ofFileMustExist,
ofNoReadOnlyReturn]
Left = 56
Top = 64
end
<>Object FontDialog1:
TFontDialog
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
Left = 136
Top = 64
end
<>Object
PrinterSetupDialog1:
TPrinterSetupDialog
Left = 176
Top = 64
end
<>Object PrintDialog1:
TPrintDialog
Left = 216
Top = 64
end
End
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
<>Object FrameForm:
TFrameForm
Left = -4
Top = -4
Width = 1032
Height = 746
Caption = 'Text editor'
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText

<b>::</b>
:: All Rights Reserved
<b>::</b>
••
<b>::</b>
:: web: http://www.qusoft.no mail:
support@qusoft.no ::
:: fax: +47 22 41 74
91 ::
}
,
unit Menu;
interface
menace
uses
SysUtils, WinTypes, WinProcs,
• • • • • • • • • • • • • • • • • • •
Messages, Classes, Graphics,
Controls,
Forms, Dialogs, StdCtrls, Quickrpt,
List, GrpList, manygrp, MD,
ExtCtrls, qrextra;
type
TMainForm = class(TForm)
GroupBox1: TGroupBox;
SimpleList: TRadioButton;
GroupedList: TRadioButton;
MasterDetail: TRadioButton;
SQLMasterDetail: TRadioButton;
Description: TMemo;
Preview: TButton;
Print: TButton;
Exit: TButton;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Image2: TImage;
RadioButton1: TRadioButton;
CompositeReport: TQuickRep;
QRCompositeReport1:
TQRCompositeReport;
<> procedure
SimpleListClick(Sender: TObject);
<pre>&lt;&gt; procedure ProviowClick(Sender: TObject);</pre>
PreviewClick(Sender: TObject);
<> procedure

<b></b>
OnClick = Cascade1Click
end
<>Object Arrangeicons1:
TMenuItem
Caption = '&Arrange icons'
OnClick = Arrangeicons1Click
end
end
end
<>Object OpenFileDialog:
TOpenDialog
Filter =
'Rich text files (*.rtf) *.rtf Plain
text files (*.txt) *.txt All' +
' files *.*'
Left = 16
Top = 24
end
end

Procedures 22		
الملف لبرنامج		
(Qr2)		
إعداد		
علاء الدين اللباد		
واتف ۷۱۳۵۷۵۶۹۰		
برنامج لصناعة التقارير		
QuickReport 2.0 for Delphi		
الملف التنفيذي ويحتوي		
{		
•••••		
•••••		
:: QuickReport 2.0 for Delphi		
1.0/2.0/3.0 ::		
::		
<b>::</b>		
:: Example reports project		
::		
::		
::		
:: Copyright (c) 1997 QuSoft AS		

begin	PrintClick(Sender: TObject);
if Report = CompositeReport then	<> procedure
begin	ExitClick(Sender: TObject);
QRCompositeReport1.Preview;	<> procedure
end else	FormActivate(Sender: TObject);
Report.Preview;	<> procedure
end;	GroupedListClick(Sender:
,	TObject);
<> procedure	<> procedure
TMainForm.PrintClick(Sender:	MasterDetailClick(Sender:
TObject);	TObject);
begin	<> procedure
if Report = CompositeReport then	SQLMasterDetailClick(Sender:
begin	TObject);
QRCompositeReport1.Print;	<> procedure
end else	RadioButton1Click(Sender:
Report.Print;	TObject);
end;	<> procedure
ciu,	QRCompositeReport1AddReports(S
<> procedure	ender: TObject);
TMainForm.ExitClick(Sender:	private
TObject);	FReport: TQuickRep;
begin	<> procedure
Close;	SetReport(Value : TQuickRep);
end;	public
ciu,	property Report : TQuickRep
<> procedure	read FReport write SetReport;
TMainForm.FormActivate(Sender:	end;
TObject);	
begin	var
if Description.Lines.Count = 0 then	MainForm: TMainForm;
SimpleListClick(Self);	
end;	implementation
ciu,	
<> procedure	{\$R *.dfm}
TMainForm.SimpleListClick(Sender	
: TObject);	<> procedure
begin	TMainForm.SetReport(Value :
Report:=ListForm.QuickRep;	TQuickRep);
end;	begin
	FReport:=Value;
<> procedure	
TMainForm.GroupedListClick(Send	Description.Lines.Assign(Report.Des
er: TObject);	cription);
begin	end;
Report := GrpListForm.QuickRep;	
end;	<> procedure
ciru,	TMainForm.PreviewClick(Sender:
	TObject);

uses	<> procedure
SysUtils, WinTypes, WinProcs,	TMainForm.MasterDetailClick(Sen
Messages, Classes, Graphics,	der: TObject);
Controls,	begin
Forms, Dialogs, quickrpt, Qrctrls,	Report:=MDForm.QuickRep;
DBTables, DB, ExtCtrls;	end;
	,
type	<> procedure
TManyGrpForm = class(TForm)	TMainForm.SQLMasterDetailClick(
QuickRep: TQuickRep;	Sender: TObject);
RepQuery: TQuery;	begin
DetailBand1: TQRBand;	Report:=
QRGroup1: TQRGroup;	ManyGrpForm.QuickRep;
QRGroup2: TQRGroup;	end;
QRDBText1: TQRDBText;	
QRDBText3: TQRDBText;	<> procedure
QRBand1: TQRBand;	TMainForm.RadioButton1Click(Sen
QRBand2: TQRBand;	der: TObject);
QRDBText4: TQRDBText;	begin
QRExpr1: TQRExpr;	Report := CompositeReport;
QRDBText5: TQRDBText;	end;
QRExpr2: TQRExpr;	
QRExpr3: TQRExpr;	<> procedure
RepQueryCustNo: TFloatField;	TMainForm.QRCompositeReport1
RepQueryCompany:	AddReports(Sender: TObject);
TStringField;	begin
RepQueryAddr1: TStringField;	with QRCompositeReport1 do
RepQueryAddr2: TStringField;	begin
RepQueryCity: TStringField;	
RepQueryState: TStringField;	Reports.Add(ListForm.QuickRep);
RepQueryZip: TStringField;	
RepQueryCountry: TStringField;	Reports.Add(GrpListForm.QuickRe
RepQueryPhone: TStringField;	<b>p)</b> ;
RepQueryFAX: TStringField;	
RepQueryTaxRate: TFloatField;	Reports.Add(MDForm.QuickRep);
RepQueryContact: TStringField;	
RepQueryLastInvoiceDate:	Reports.Add(ManyGrpForm.Quick
TDateTimeField;	Rep);
RepQueryOrderNo: TFloatField;	end;
RepQueryCustNo 1: TFloatField;	end;
RepQuerySaleDate:	_
TDateTimeField;	end.
RepQueryShipDate:	
TDateTimeField;	111111111111111111111111111111111111111
RepQueryEmpNo: TIntegerField;	2
RepQueryShipToContact:	unit Manygrp;
TStringField;	
RepQueryShipToAddr1:	interface

nwivata
private { Private declarations }
public (Public dealerations)
{ Public declarations }
end;
var
ManyGrpForm: TManyGrpForm;
• • •
implementation
(OD * 10 )
{\$R *.dfm}
end.
2
unit Md;
• , 6
interface
uses
SysUtils, WinTypes, WinProcs,
Messages, Classes, Graphics,
Controls,
Forms, Dialogs, ExtCtrls, quickrpt,
DB, DBTables, Qrctrls;
tymo
type TMDForm = class(TForm)
, ,
QuickRep: TQuickRep; DetailBand1: TQRBand;
Detaildaligi: IQKdalig;
ColumnHeaderBand1: TQRBand;
ColumnHeaderBand1: TQRBand; PageFooterBand1: TQRBand;
ColumnHeaderBand1: TQRBand; PageFooterBand1: TQRBand; TitleBand1: TQRBand;
ColumnHeaderBand1: TQRBand; PageFooterBand1: TQRBand; TitleBand1: TQRBand; Customer: TTable;
ColumnHeaderBand1: TQRBand; PageFooterBand1: TQRBand; TitleBand1: TQRBand; Customer: TTable; QRDBText1: TQRDBText;
ColumnHeaderBand1: TQRBand; PageFooterBand1: TQRBand; TitleBand1: TQRBand; Customer: TTable; QRDBText1: TQRDBText; QRLabel1: TQRLabel;
ColumnHeaderBand1: TQRBand; PageFooterBand1: TQRBand; TitleBand1: TQRBand; Customer: TTable; QRDBText1: TQRDBText; QRLabel1: TQRLabel; QRDBText2: TQRDBText;
ColumnHeaderBand1: TQRBand; PageFooterBand1: TQRBand; TitleBand1: TQRBand; Customer: TTable; QRDBText1: TQRDBText; QRLabel1: TQRLabel; QRDBText2: TQRDBText; QRLabel2: TQRLabel;
ColumnHeaderBand1: TQRBand; PageFooterBand1: TQRBand; TitleBand1: TQRBand; Customer: TTable; QRDBText1: TQRDBText; QRLabel1: TQRLabel; QRDBText2: TQRDBText; QRLabel2: TQRDBText; QRDBText3: TQRDBText;
ColumnHeaderBand1: TQRBand; PageFooterBand1: TQRBand; TitleBand1: TQRBand; Customer: TTable; QRDBText1: TQRDBText; QRLabel1: TQRLabel; QRDBText2: TQRDBText; QRLabel2: TQRDBText; QRLabel3: TQRDBText;
ColumnHeaderBand1: TQRBand; PageFooterBand1: TQRBand; TitleBand1: TQRBand; Customer: TTable; QRDBText1: TQRDBText; QRLabel1: TQRLabel; QRDBText2: TQRDBText; QRLabel2: TQRLabel; QRDBText3: TQRDBText; QRLabel3: TQRDBText; QRLabel3: TQRDBText;
ColumnHeaderBand1: TQRBand; PageFooterBand1: TQRBand; TitleBand1: TQRBand; Customer: TTable; QRDBText1: TQRDBText; QRLabel1: TQRLabel; QRDBText2: TQRDBText; QRLabel2: TQRLabel; QRDBText3: TQRDBText; QRLabel3: TQRLabel; QRDBText4: TQRDBText; QRLabel4: TQRDBText;
ColumnHeaderBand1: TQRBand; PageFooterBand1: TQRBand; TitleBand1: TQRBand; Customer: TTable; QRDBText1: TQRDBText; QRLabel1: TQRLabel; QRDBText2: TQRDBText; QRLabel2: TQRLabel; QRDBText3: TQRDBText; QRLabel3: TQRDBText; QRLabel4: TQRDBText; QRLabel5: TQRDBText;
ColumnHeaderBand1: TQRBand; PageFooterBand1: TQRBand; TitleBand1: TQRBand; Customer: TTable; QRDBText1: TQRDBText; QRLabel1: TQRLabel; QRDBText2: TQRDBText; QRLabel2: TQRLabel; QRDBText3: TQRDBText; QRLabel3: TQRLabel; QRDBText4: TQRDBText; QRLabel4: TQRLabel; QRLabel5: TQRLabel; QRLabel5: TQRLabel;
ColumnHeaderBand1: TQRBand; PageFooterBand1: TQRBand; TitleBand1: TQRBand; Customer: TTable; QRDBText1: TQRDBText; QRLabel1: TQRLabel; QRDBText2: TQRDBText; QRLabel2: TQRLabel; QRDBText3: TQRDBText; QRLabel3: TQRLabel; QRDBText4: TQRDBText; QRLabel4: TQRDBText; QRLabel5: TQRLabel; QRLabel5: TQRLabel; QRLabel6: TQRLabel;
ColumnHeaderBand1: TQRBand; PageFooterBand1: TQRBand; TitleBand1: TQRBand; Customer: TTable; QRDBText1: TQRDBText; QRLabel1: TQRLabel; QRDBText2: TQRDBText; QRLabel2: TQRLabel; QRDBText3: TQRDBText; QRLabel3: TQRLabel; QRDBText4: TQRDBText; QRLabel4: TQRLabel; QRLabel5: TQRLabel; QRLabel5: TQRLabel;

TStringField;
RepQueryShipToAddr2:
TStringField;
RepQueryShipToCity:
TStringField;
RepQueryShipToState:
TStringField;
RepQueryShipToZip:
TStringField;
RepQueryShipToCountry:
TStringField;
RepQueryShipToPhone:
TStringField;
RepQueryShipVIA: TStringField;
RepQueryPO: TStringField;
RepQueryTerms: TStringField;
RepQueryPaymentMethod:
TStringField;
RepQueryItemsTotal:
TCurrencyField;
RepQueryTaxRate_1:
TFloatField;
RepQueryFreight:
TCurrencyField;
RepQueryAmountPaid:
TCurrencyField;
RepQueryOrderNo_1:
TFloatField;
RepQueryItemNo: TFloatField;
RepQueryPartNo: TFloatField;
RepQueryQty: TIntegerField;
RepQueryDiscount: TFloatField;
RepQueryPartNo_1: TFloatField;
RepQueryVendorNo:
TFloatField;
RepQueryDescription:
TStringField;
RepQueryOnHand: TFloatField;
RepQueryOnOrder: TFloatField;
RepQueryCost: TCurrencyField;
RepQueryListPrice:
TCurrencyField;
QRLabel1: TQRLabel;
QRLabel2: TQRLabel;
QRLabel3: TQRLabel;
QRLabel4: TQRLabel;
QRExpr4: TQRExpr;
QRLabel5: TQRLabel;
ChildBand1: TQRChildBand;

TitleBand1: TQRBand;
Customer: TTable;
QRDBText1: TQRDBText;
QRLabel1: TQRLabel;
QRDBText2: TQRDBText;
QRLabel2: TQRLabel;
QRDBText3: TQRDBText;
QRLabel3: TQRLabel;
QRDBText4: TQRDBText;
QRLabel4: TQRLabel;
,
QRLabel5: TQRLabel;
QRDBText5: TQRDBText;
QRLabel6: TQRLabel;
QRSysData1: TQRSysData;
QRSysData2: TQRSysData;
QRExpr1: TQRExpr;
QRGroup1: TQRGroup;
private
{ Private declarations }
public
{ Public declarations }
end;
var
var GrnListForm: TGrnListForm:
var GrpListForm: TGrpListForm;
GrpListForm: TGrpListForm;
1 11
GrpListForm: TGrpListForm; implementation
GrpListForm: TGrpListForm;
GrpListForm: TGrpListForm; implementation  {\$R *.dfm}
GrpListForm: TGrpListForm; implementation  {\$R *.dfm} end.
GrpListForm: TGrpListForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
GrpListForm: TGrpListForm;  implementation  {\$R *.dfm}  end.
GrpListForm: TGrpListForm;  implementation  {\$R *.dfm}  end.
GrpListForm: TGrpListForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
GrpListForm: TGrpListForm; implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
GrpListForm: TGrpListForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
GrpListForm: TGrpListForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
GrpListForm: TGrpListForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
GrpListForm: TGrpListForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
GrpListForm: TGrpListForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
GrpListForm: TGrpListForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
GrpListForm: TGrpListForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
GrpListForm: TGrpListForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
GrpListForm: TGrpListForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
GrpListForm: TGrpListForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

Orders: TTable;
OIUCID, II anic,
CustomerDS: TDataSource;
QRSubDetail1: TQRSubDetail;
QRDBText6: TQRDBText;
QRDBText7: TQRDBText;
QRDBText10: TQRDBText;
QRLabel7: TQRLabel;
QRLabel8: TQRLabel;
QRLabel9: TQRLabel;
GroupFooterBand1: TQRBand;
QRExpr1: TQRExpr;
QRLabel10: TQRLabel;
QRLabel11: TQRLabel;
QRExpr2: TQRExpr;
ChildBand1: TQRChildBand;
private
{ Private declarations }
public (Dublic dealonations)
{ Public declarations }
end;
var
MDForm: TMDForm;
implementation
{\$R *.dfm}
end.
viiw!
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
V == VEV
111111111111111111111111111111111111111
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
unit Grplist;  interface  uses
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
unit Grplist; interface uses SysUtils, WinTypes, WinProcs,
unit Grplist;  interface  uses  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
unit Grplist;  interface  uses  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, quickrpt,
unit Grplist;  interface  uses  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
unit Grplist;  interface  uses  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, quickrpt, DB, DBTables, Qrctrls;
unit Grplist;  interface  uses  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, quickrpt, DB, DBTables, Qrctrls;  type
unit Grplist;  interface  uses  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, quickrpt, DB, DBTables, Qrctrls;  type  TGrpListForm = class(TForm)
unit Grplist;  interface  uses  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, quickrpt, DB, DBTables, Qrctrls;  type  TGrpListForm = class(TForm) QuickRep: TQuickRep;
unit Grplist;  interface  uses  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, quickrpt, DB, DBTables, Qrctrls;  type  TGrpListForm = class(TForm) QuickRep: TQuickRep; DetailBand1: TQRBand;
unit Grplist;  interface  uses  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, quickrpt, DB, DBTables, Qrctrls;  type  TGrpListForm = class(TForm) QuickRep: TQuickRep;

Top = 89
Width = 787
Height = 618
Caption = 'MDForm'
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -13
Font.Name = 'System'
Font.Style = []
OldCreateOrder = True
Scaled = False
PixelsPerInch = 96
TextHeight = 16
<> Object Customer: TTable
Active = True
DatabaseName = 'DBDEMOS'
IndexFieldNames = 'Company'
ReadOnly = True
TableName = 'CUSTOMER.DB'
Left = 20
Top = 76
end
<> Object Orders: TTable
Active = True
DatabaseName = 'DBDEMOS'
IndexFieldNames = 'CustNo'
MasterFields = 'CustNo'
MasterSource = CustomerDS
ReadOnly = True
TableName = 'ORDERS.DB'
Left = 12
Top = 132
end
<> Object CustomerDS:
<b>TDataSource</b>
DataSet = Customer
Left = 28
Top = 20
end
End
2
<> Object ManyGrpForm:
TManyGrpForm
Left = 135
Top = 87
Toh Ol

ColumnHeaderBand1: TQRBand; PageFooterBand1: TQRBand; TitleBand1: TQRBand; Customer: TTable; QRDBText1: TQRDBText; QRLabel1: TQRLabel;
PageFooterBand1: TQRBand; TitleBand1: TQRBand; Customer: TTable; QRDBText1: TQRDBText;
TitleBand1: TQRBand; Customer: TTable; QRDBText1: TQRDBText;
Customer: TTable; QRDBText1: TQRDBText;
ODI abali, TODI abali
QKLabeii. TQKLabei,
QRDBText2: TQRDBText;
QRLabel2: TQRLabel;
QRDBText3: TQRDBText;
QRLabel3: TQRLabel;
QRDBText4: TQRDBText;
QRLabel4: TQRLabel;
QRLabel5: TQRLabel;
QRDBText5: TQRDBText;
QRLabel6: TQRLabel;
QRSysData1: TQRSysData;
QRSysData2: TQRSysData;
private
{ Private declarations }
public
{ Public declarations }
end;
var
ListForm: TListForm;
implementation
{\$R *.dfm}
and a
end.

<> Objects 72
الملف النصي لبرنامج
(Qr2)
إعداد
علاء الدين اللباد
واتف ۷۱۳۷۱۹۰۱
1
<> Object MDForm:
TMDForm
Left = 117

Size = 15
end
<> Object RepQueryState:
TStringField
FieldName = 'State'
end
<> Object RepQueryZip:
TStringField
FieldName = 'Zip'
Size = 10
end
<> Object
RepQueryCountry: TStringField
FieldName = 'Country'
end
<> Object RepQueryPhone:
TStringField
FieldName = 'Phone'
Size = 15
end
<> Object RepQueryFAX:
TStringField
FieldName = 'FAX'
Size = 15
end
<> Object
RepQueryTaxRate: TFloatField
FieldName = 'TaxRate'
end
<> Object RepQueryContact:
TStringField
FieldName = 'Contact'
end
<> Object
RepQueryLastInvoiceDate:
TDateTimeField
FieldName = 'LastInvoiceDate'
end
<> Object
RepQueryOrderNo: TFloatField
FieldName = 'OrderNo'
end Object
<> Object  PenOveryCustNo. 1: TFlootField
RepQueryCustNo_1: TFloatField
FieldName = 'CustNo_1'
end
<> Object
RepQuerySaleDate:
TDateTimeField

Width = 770
Height = 657
Caption = 'ManyGrpForm'
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -13
Font.Name = 'System'
Font.Style = []
OldCreateOrder = True
Scaled = False
PixelsPerInch = 96
TextHeight = 16
<> Object RepQuery: TQuery
Active = True
DatabaseName = 'DBDEMOS'
SQL.Strings = (
'select * from customer a, orders
b, items c, parts d'
'where a.custno = b.custno'
' and b.orderno = c.orderno'
' and c.partno = d.partno'
'order by a.company, orderno')
Left = 18
Top = 10
<> Object RepQueryCustNo:
TFloatField
FieldName = 'CustNo'
end
<> Object
RepQueryCompany: TStringField
FieldName = 'Company'
Size = 30
end
<> Object RepQueryAddr1:
TStringField
FieldName = 'Addr1'
Size = 30
end
<pre>&lt;&gt; Object RepQueryAddr2: TStringField</pre>
TStringField FieldName = !Addv2!
FieldName = 'Addr2'
Size = 30
end City
<> Object RepQueryCity:
TStringField
FieldName = 'City'

TStringField
FieldName = 'ShipToPhone'
Size = 15
end
<> Object
RepQueryShipVIA: TStringField
FieldName = 'ShipVIA'
Size = 7
end
<> Object RepQueryPO:
TStringField
FieldName = 'PO'
Size = 15
end  Object Pen Overy Terms
<> Object RepQueryTerms:
TStringField  FieldName = 'Towns'
FieldName = 'Terms'
Size = 6
end
<> Object
RepQueryPaymentMethod:
TStringField
FieldName = 'PaymentMethod'
Size = 7
end
<> Object
RepQueryItemsTotal:
TCurrencyField
FieldName = 'ItemsTotal'
end
<> Object
RepQueryTaxRate_1: TFloatField
FieldName = 'TaxRate_1'
end
<> Object RepQueryFreight:
TCurrencyField
FieldName = 'Freight'
end
<> Object
RepQueryAmountPaid:
TCurrencyField
FieldName = 'AmountPaid'
end
<> Object
RepQueryOrderNo 1: TFloatField
FieldName = 'OrderNo 1'
end
<> Object RepQueryItemNo:
TFloatField
1 1 Watt Kiu

FieldName = 'SaleDate'
end
<> Object
RepQueryShipDate:
<b>TDateTimeField</b>
FieldName = 'ShipDate'
end
<> Object RepQueryEmpNo:
TIntegerField
FieldName = 'EmpNo'
end
<> Object
RepQueryShipToContact:
TStringField
FieldName = 'ShipToContact'
end
<> Object
RepQueryShipToAddr1:
TStringField
FieldName = 'ShipToAddr1'
Size = 30
end
<> Object
RepQueryShipToAddr2:
TStringField
FieldName = 'ShipToAddr2'
Size = 30
end
<> Object
RepQueryShipToCity: TStringField
FieldName = 'ShipToCity'
Size = 15
end
<> Object
RepQueryShipToState: TStringField
FieldName = 'ShipToState'
end
<> Object
=
RepQueryShipToZip: TStringField
FieldName = 'ShipToZip'
Size = 10
end
<> Object
RepQueryShipToCountry:
TStringField
FieldName = 'ShipToCountry'
end
<> Object
RepQueryShipToPhone:

TMainForm
Left = 182
Top = 121
BorderStyle = bsDialog
Caption = 'QuickReport 2.0
Example reports'
ClientHeight = 331 ClientWidth = 552
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
Scaled = False
OnActivate = FormActivate
PixelsPerInch = 96
TextHeight = 16
<> Object Label1: TLabel
Left = 56
Top = 4
Width = 174
Height = 34
Caption = 'QuickReport'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -29
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
Transparent = True
end
<> Object Label2: TLabel
Left = 236
Top = 18
Width = 70
Height = 16
Caption = 'Version 2.0'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'Arial'
Font.Style = [fsBold]

FieldName = 'ItemNo'
end
<> Object RepQueryPartNo:
TFloatField
FieldName = 'PartNo'
end
<> Object RepQueryQty:
TIntegerField
FieldName = 'Qty'
end
<> Object
RepQueryDiscount: TFloatField
FieldName = 'Discount'
end
<> Object
RepQueryPartNo 1: TFloatField
FieldName = 'PartNo 1'
end
<> Object
RepQueryVendorNo: TFloatField
FieldName = 'VendorNo'
end
<> Object
RepQueryDescription: TStringField
FieldName = 'Description'
Size = 30
end
<> Object
RepQueryOnHand: TFloatField
FieldName = 'OnHand'
end
<> Object
RepQueryOnOrder: TFloatField
FieldName = 'OnOrder'
end
<> Object RepQueryCost:
TCurrencyField
FieldName = 'Cost'
end
<> Object
RepQueryListPrice:
TCurrencyField
FieldName = 'ListPrice'
end
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
3
<> Object MainForm:
Object Maint of III.

Width = 189
Height = 17
Caption = 'Simple list with
group'
TabOrder = 1
OnClick = GroupedListClick
end
<> Object MasterDetail:
TRadioButton
Left = 12
Top = 67
Width = 193
Height = 17
Caption = 'Master/detail'
TabOrder = 2
OnClick = MasterDetailClick
end
<> Object SQLMasterDetail:
TRadioButton
Left = 12
Top = 91
Width = 201
Height = 17
Caption = 'Several grouping
levels'
TabOrder = 3
OnClick =
<b>SQLMasterDetailClick</b>
end
<> Object Description:
TMemo
Left = 220
Top = 16
Width = 313
Height = 201
TabStop = False
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
ReadOnly = True
TabOrder = 4
end
<> Object RadioButton1:
Object RadioDuttonii.

ParentFont = False
Transparent = True
end
<> Object Label3: TLabel
Left = 56
Top = 36
Width = 333
Height = 16
Caption = 'Copyright 1995 - 1997
QuSoft AS http://www.qusoft.com'
Transparent = True
end
<> Object Image2: TImage
Left = 6
Top = 6
Width = 47
Height = 47
Picture.Data = {
C0C0C0C0C0C0C0C0C0C0C0C0C
0C0}
end
<> Object GroupBox1:
TGroupBox
Left = 4
Top = 58
Width = 541
Height = 225
Caption = 'Select report'
Color = clBtnFace
ParentColor = False
TabOrder = 0
<> Object SimpleList:
TRadioButton
Left = 12
Top = 20
Width = 193
Height = 17
Caption = 'Simple list'
Checked = True
TabOrder = 0
TabStop = True
OnClick = SimpleListClick
end
<> Object GroupedList:
TRadioButton
Left = 12
Top = 44
- T

Caption = 'ListForm'
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -13
Font.Name = 'System'
Font.Style = []
OldCreateOrder = True
Scaled = False
PixelsPerInch = 96
TextHeight = 16
<> Object Customer: TTable
Active = True
DatabaseName = 'DBDEMOS'
IndexFieldNames = 'Company'
ReadOnly = True
TableName = 'CUSTOMER.DB'
Left = 4
<b>Top</b> = 4
end
End
- Ditt
111111111111111111111111111111111111111
5
<> Object GrpListForm:
TGrpListForm
Left = 138
Top = 108
Width = 680
Height = 587
Caption = 'GrpListForm'
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -13
Font.Name = 'System'
Font.Style = []
1
OldCreateOrder = True
Scaled = False
Scaled = False PixelsPerInch = 96
Scaled = False PixelsPerInch = 96 TextHeight = 16
Scaled = False PixelsPerInch = 96 TextHeight = 16 <> Object Customer: TTable
Scaled = False PixelsPerInch = 96 TextHeight = 16
Scaled = False PixelsPerInch = 96 TextHeight = 16 <> Object Customer: TTable
Scaled = False PixelsPerInch = 96 TextHeight = 16 <> Object Customer: TTable Active = True
Scaled = False PixelsPerInch = 96 TextHeight = 16 <> Object Customer: TTable Active = True DatabaseName = 'DBDEMOS'

TRadioButton
Left = 12
<b>Top = 114</b>
Width = 179
Height = 17
Caption = 'Composite report'
TabOrder = 5
OnClick = RadioButton1Click
end
end
<> Object Preview: TButton
Left = 4
Top = 292
Width = 89
Height = 29
Caption = 'P&review'
TabOrder = 1
OnClick = PreviewClick end
<> Object Print: TButton
Left = 100
Top = 292
Width = 89
Height = 29
Caption = '&Print'
TabOrder = 2
OnClick = PrintClick
end
<> Object Exit: TButton
Left = 456
Top = 292
<b>Width = 89</b>
Height = 29
Caption = 'E&xit'
TabOrder = 3
OnClick = ExitClick
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
5
<> Object ListForm:
TListForm
Left = 138
Top = 115
Width = 685
Width = 685 Height = 587
11cigii - 30/

ExtCtrls, StdCtrls, quickrpt, Db,
DBTables, printers, qrextra,
QRExport,
qrprntr;
const
Composite_Description = 'The
composite control can be used to link
several'+
' reports together as a
single report.';
g · · · · · · · · · · · · · · · · · · ·
type
TfrmQR3Demo = class(TForm)
QRTextFilter1: TQRTextFilter;
QRCSVFilter1: TQRCSVFilter;
ORHTMLFilter1:
TORHTMLFilter;
/
QRCompositeReport1:
TQRCompositeReport;
Label1: TLabel;
VersionLbl: TLabel;
Label3: TLabel;
GroupBox1: TGroupBox;
rbCreateList: TRadioButton;
btnPreview: TButton;
btnPrint: TButton;
rbMasterDetail: TRadioButton;
Description: TMemo;
OpenDialog1: TOpenDialog;
cbPreview: TComboBox;
Label4: TLabel;
rbExprMemo: TRadioButton;
rbImage: TRadioButton;
rbBasicMD: TRadioButton;
btnFilters: TButton;
rbComposite: TRadioButton;
Label5: TLabel;
rbNeedData: TRadioButton;
rbFormLetter: TRadioButton;
btnExport: TButton;
SaveDialog1: TSaveDialog;
rbAbout: TRadioButton;
rbGrouping: TRadioButton;
Image2: TImage;
<> procedure
CreateList(Sender: TObject);
<> procedure

TableName = 'CUSTOMER.DB'
Left = 12
Top = 12
end
End
111111111111111111111111111111111111111

الملف التنفيذي لبرنامج		
(Qr3)		
إعداد		
A1 .111 .1		
علاء الدين اللباد		
واتف ۷۱۳۵۷۵۶۹۰		
علاء الدين اللباد علاء الدين اللباد		
procedures 110		
:: QuickReport 3.0 for Delphi		
3.0/4.0/5.0/6.0 ::		
::		
::		
:: Example reports project		
:		
::		
:: Copyright (c) 1995-1999 QuSoft		
AS ::		
:: All Rights Reserved		
::		
.: 		
:: web: http://www.qusoft.com fax:		
+47 22 41 74 91 ::		
:::::::::::}		
unit mdmain;		
intoufogo		
interface		
uses		
Windows, Messages, SysUtils,		
Classes, Graphics, Controls, Forms,		
Dialogs,		

TCustomQuiskDon wood EDonout	Г
TCustomQuickRep read FReport	
write SetReport;	-
end;	
	_
var	
frmQR3Demo: TfrmQR3Demo;	
	-
implementation	
uses mdrpt, exprmemo, image,	-
basicmd, needdata, frmltr,	
history, grouping;	
{\$R *.dfm}	
<> procedure	-
TfrmQR3Demo.SetReport(Value:	
TCustomQuickRep);	
begin	H
FReport := Value;	
if Value $\Leftrightarrow$ nil then	H
if Value =	
TCustomQuickRep(QRCompositeR	F
eport1) then	
Description.Lines.Text :=	-
Composite_Description	
else	
	F
Description.Lines.Assign(Report.Des	
cription);	H
end;	
Z S nucceduuc	
<> procedure TfrmOP3Domo Croatal ist(Sandar)	F
TfrmQR3Demo.CreateList(Sender:	
TObject);	
aReport : TCustomQuickRep;	
SomeFields: TStringList;	Ī
9 /	
MyTable: TTable; nIdx: integer;	ľ
3 /	ľ
Create a table on the fly this	F
{ Create a table on the fly, this	r
example uses a table from the demo database }	
MyTable := TTable.Create(self);	
wry rable :- 1 rable. Create(self);	
( anada tha list of Calarta autor t	F
{ create the list of fields to output	F
from the table }	F
SomeFields := TStringList.Create;	L

QRCompositeReport1AddReports(S
ender: TObject); <> procedure
btnCRClick(Sender: TObject);
<> procedure
rbCreateListClick(Sender:
TObject);
<> procedure
rbMasterDetailClick(Sender:
TObject);
<> procedure
FormCreate(Sender: TObject);
<> procedure
FormActivate(Sender: TObject); <> procedure
btnPreviewClick(Sender: TObject);
<> procedure
rbExprMemoClick(Sender:
TObject);
<> procedure
rbImageClick(Sender: TObject);
<> procedure
rbBasicMDClick(Sender: TObject);
<> procedure
btnFiltersClick(Sender: TObject);
<> procedure
rbCompositeClick(Sender:
TObject);
<> procedure
rbNeedDataClick(Sender: TObject);
<> procedure
rbFormLetterClick(Sender:
TObject);
<> procedure
btnExportClick(Sender: TObject);
<> procedure
rbAboutClick(Sender: TObject);
<> procedure
rbGroupingClick(Sender: TObject);
private
{ Private declarations }
FReport: TCustomQuickRep;
CreateListReport : TQuickRep;
<> procedure
SetReport(Value :
TCustomQuickRep);
public
{ Public declarations }
property Report :

// won out	with MwTable de
// report.	with MyTable do
C 11 04	begin
for nIdx := 0 to	DatabaseName := 'DBDEMOS';
aReport.Bands.ColumnHeaderBand.	TableName := 'COUNTRY.DB';
ControlCount -1 do	ReadOnly := True;
if D. J. G. J. W. J. D. J.	Active := True;
aReport.Bands.ColumnHeaderBand.	
Controls[nIdx] is TQRPrintable	// For this example, we will pull
then	the field names from the table
with	// If you wanted to only use some
TQRPrintable(aReport.Bands.Colu	of the fields, you would edit
mnHeaderBand.Controls[nIdx]) do	// this list.
Left := Left - (5 * nIdx);	for nIdx := 0 to FieldCount - 1 do
for alds to 0.40	
for nIdx := 0 to	SomeFields.Add(Fields[nIdx].FieldN
aReport.Bands.DetailBand.ControlC	ame);
ount -1 do	end;
if	
aReport.Bands.DetailBand.Controls[	// If you didn't create the report,
nIdx] is TQRPrintable then	you must set the report object to nil
with TOP Prints bloke Demont Panda Detail	// before calling QRCreateList
TQRPrintable(aReport.Bands.Detail	
Band.Controls[nIdx]) do	areport := nil;
Left := Left - (5 * nIdx);	
	{ Build the report }
{ You can change the report objects	// 70
before calling the report }	// If you change the displaywidth, it
// areport.page.orientation :=	will be reflecte in the created
poLandscape;	// report
{preview or print the report}	
{preview or print the report}	with MyTable.Fields[1] do
if sender = btnPreview then	DisplayWidth := DisplayWidth
begin	div 2;
case cbPreview.ItemIndex of	44
0: areport.preview;	// create the report
	QRCreateList(aReport, nil,
1: areport.previewModal;	MyTable, 'Country Listing',
2: areport.previewModeless;	SomeFields);
end;	
end	// Make the column header's font
else if sender = btnPrint then	use bold attribute
areport.print;	
(all dama for the 11 to )	aReport.Bands.ColumnHeaderBand.
{ all done, free the objects }	Font.Style := [fsBold];
aReport.Free;	
MyTable.Free;	// Now adjust the spacing of the
SomeFields.Free;	fields. There isn't any reason to
end;	// do this, this is just to show how to
	access the controls on the

if i >=0 then	<> procedure
VersionLbl.Caption := 'Version' +	TfrmQR3Demo.QRCompositeRepor
copy(cQRName, i, 99);	t1AddReports(Sender: TObject);
// Create the report object used by	begin
the QRCreateList() function	// The OnAddReports event is
CreateListReport :=	called by the CompositeReport
TQuickRep.Create(self);	// to add the reports to list of
CreateListReport.Description.Text	reports
:= 'Example of how to call the	with
QRCreateList function';	QRCompositeReport1.Reports do
cbPreview.ItemIndex := 0;	begin
end;	
	Add(frmMasterDetail.QuickRep1);
<> procedure	Add(frmBasicMD.QuickRep1);
TfrmQR3Demo.FormActivate(Send	Add(frmImageRpt.QuickRep1);
er: TObject);	end;
begin	end;
if Description.Lines.Count = 0 then	
begin	<> procedure
rbCreateListClick(Self);	TfrmQR3Demo.btnCRClick(Sender
rbCreateList.Checked := True;	: TObject);
end;	begin
end;	QRCompositeReport1.Preview;
onu,	end;
<> procedure	onu,
TfrmQR3Demo.btnPreviewClick(Se	<> procedure
nder: TObject);	TfrmQR3Demo.rbCreateListClick(S
begin	ender: TObject);
// This code is more complicated	begin
than what you would	Report := CreateListReport;
// typically need to run a report.	end;
Most of this code	chu,
// is to handle the selection of the	<> procedure
various types of	TfrmQR3Demo.rbMasterDetailClic
// reports.	k(Sender: TObject);
// Icports.	begin
if report <> nil then	Report :=
begin	frmMasterDetail.QuickRep1;
9	end;
if report = CreateListReport then	enu,
CreateList(Sender)	/ >nunnaduun
else if report = TCystomOvielPor(OPCompositeP	<pre>&lt;&gt; procedure TfrmOP3Domo FormCroats(Sandar)</pre>
TCustomQuickRep(QRCompositeR	TfrmQR3Demo.FormCreate(Sender
eport1) then	: TObject);
begin	var i: integer;
if sender = btnPreview then	begin
QRCompositeReport1.preview	// Get the current QuickReport
else	version number from
QRCompositeReport1.print;	// the qrprntr unit
end	i := pos(' ', cQRName);

end;	else
	begin
<> procedure	if sender = btnPreview then
TfrmQR3Demo.rbCompositeClick(S	begin
ender: TObject);	case cbPreview.ItemIndex of
begin	0: report.preview;
Report :=	1: report.previewModal;
TCustomQuickRep(QRCompositeR	2: report.previewModeless;
eport1);	end;
end;	end end
chu,	
<> procedure	else if sender = btnPrint then
TfrmQR3Demo.rbNeedDataClick(S	report.print;
ender: TObject);	end;
begin	end;
Report :=	end;
frmNeedData.QuickRep1;	
end;	<> procedure
enu;	TfrmQR3Demo.rbExprMemoClick(
/ > nwooduwo	Sender: TObject);
<pre>&lt;&gt; procedure Tfrm OB 2D one on the count of the Click()</pre>	begin
TfrmQR3Demo.rbFormLetterClick(	Report :=
Sender: TObject);	frmExprmemo.QuickRep1;
begin	end;
Report :=	
frmFormLetter.QuickRep1;	<> procedure
end;	TfrmQR3Demo.rbImageClick(Send
(/70) 6 1)	er: TObject);
// The following code show how to	begin
explicitly call an export	Report :=
// filter without going through the	frmImageRpt.QuickRep1;
preview	end;
<> procedure	
TfrmQR3Demo.btnExportClick(Sen	<> procedure
der: TObject);	TfrmQR3Demo.rbBasicMDClick(Se
begin	nder: TObject);
btnExport.Enabled := False;	begin
with SaveDialog1 do	Report :=
begin	frmBasicMD.QuickRep1;
if Execute then	end;
begin	
	<> procedure
frmFormLetter.QuickRep1.ExportT	TfrmQR3Demo.btnFiltersClick(Sen
oFilter(TQRCommaSeparatedFilter.	der: TObject);
Create(FileName));	begin
{	MessageDlg('When an export filter
Other filters:	control is dropped on a form, it''s
HTML: TQRHTMLDocumentFilter	added to all of the
ASCII: TQRAsciiExportFilter	previews',mtInformation, [mbok],
CSV: TQRCommaSeparatedFilter	0);
	- 17

::::::}
unit basicmd;
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
Db, DBTables, ExtCtrls, QuickRpt,
Qretrls;
Q1 ctris,
type
TfrmBasicMD = class(TForm)
QuickRep1: TQuickRep;
dsCustomer: TDataSource;
,
tbCustomer: TTable;
DetailBand1: TQRBand;
QRSubDetail1: TQRSubDetail;
QRDBText1: TQRDBText;
QRDBText2: TQRDBText;
qryOrders: TQuery;
QRGroup1: TQRGroup;
QRDBText3: TQRDBText;
ColumnHeaderBand1: TQRBand;
QRLabel1: TQRLabel;
QRLabel2: TQRLabel;
QRLabel3: TQRLabel;
QRDBText4: TQRDBText;
QRLabel4: TQRLabel;
QRBand1: TQRBand;
QRExpr1: TQRExpr;
QRLabel5: TQRLabel;
private
{ Private declarations }
public
{ Public declarations }
end;
,
var
frmBasicMD: TfrmBasicMD;
,
implementation
{\$R *.dfm}
(Ψ12 ·WIIII)

Г
In Professional Version:
RTF: TQRRTFExportFilter
WMF: TQRWMFExportFilter
Excel: TQRXLSFilter
}
end;
end;
btnExport.Enabled := True;
end;
,
<> procedure
TfrmQR3Demo.rbAboutClick(Send
er: TObject);
begin
Report := frmHistory.QuickRep1;
end;
<> procedure
TfrmQR3Demo.rbGroupingClick(Se
nder: TObject);
begin
Report :=
frmGrouping.QuickRep1;
end;
enu;
end.
enu.
2
{
•••••
:: QuickReport 3.0 for Delphi
3.0/4.0/5.0 ::
::
::
:: Basic Master/Detail report with
group bands ::
Stoup builds
••
••
::
:: :: Copyright (c) 1995-1999 QuSoft AS ::
AS ::
10 0
AS ::
AS ::
AS :: :: All Rights Reserved :: ::
AS ::

4h Cyasta ya ay Cityy TCtyrin a Fiold.
tbCustomerCity: TStringField;
tbCustomerState: TStringField;
tbCustomerZip: TStringField;
tbCustomerCountry:
TStringField;
tbCustomerPhone: TStringField;
tbCustomerFAX: TStringField;
tbCustomerTaxRate: TFloatField;
tbCustomerContact:
TStringField;
tbCustomerLastInvoiceDate:
TDateTimeField;
QRExprMemo1:
TQRExprMemo;
private
{ Private declarations }
public
{ Public declarations }
,
end;
var
frmExprmemo: TfrmExprmemo;
implementation
implementation
Implementation
{\$R *.dfm}
{\$R *.dfm}
{\$R *.dfm} end.
{\$R *.dfm}  end.
{\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
{\$R *.dfm}  end.
{\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
{\$R *.dfm}  end.
{\$R *.dfm}  end.
{\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
{\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
{\$R *.dfm}  end.

end.
\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
3
{
••••••
:: QuickReport 3.0 for Delphi
3.0/4.0/5.0 ::
::
::
:: Demonstration on how to use the
TQRExprMemo control ::
::
::
:: Copyright (c) 1995-1999 QuSoft
AS ::
:: All Rights Reserved
::
::
<b>::</b>
:: web: http://www.qusoft.com fax:
+47 22 41 74 91 ::
<b>::::::</b> }
unit exprmemo;
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
QuickRpt, Qrctrls, Db, DBTables,
ExtCtrls;
type
TfrmExprmemo = class(TForm)
QuickRep1: TQuickRep;
tbCustomer: TTable;
DetailBand1: TQRBand;
tbCustomerCustNo: TFloatField;
tbCustomerCompany:
TStringField;
tbCustomerAddr1: TStringField;
tbCustomerAddr2: TStringField;

var	
frmFormLetter: TfrmFormLetter;	
	_
implementation	4
(07) 1 10	_
{\$R *.dfm}	
	4
<> procedure	
TfrmFormLetter.DetailBand1Before	
Print(Sender: TQRCustomBand;	-
var PrintBand: Boolean);	-
begin if FirstDetail then	-
FirstDetail := False	-
else	-
QuickRep1.NewPage;	-
end;	-
cnu,	-
<> procedure	
TfrmFormLetter.QuickRep1Before	
Print(Sender: TCustomQuickRep;	
var PrintReport: Boolean);	1
begin	
FirstDetail := True;	
end;	
end.	
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	_
5	_
	_
{	
:: QuickReport 3.0 for Delphi	1
3.0/4.0/5.0 ::	
••	1
::	
:: Simple report for grouping data	
::	
::	
::	
:: Copyright (c) 1995-1999 QuSoft	
AS ::	
:: All Rights Reserved	
::	
<b>::</b>	
::	
:: web: http://www.qusoft.com fax:	

············}
unit frmltr;
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
Db, DBTables, ExtCtrls, QuickRpt,
Qrctrls;
type
TfrmFormLetter = class(TForm)
QuickRep1: TQuickRep;
qryEmployee: TQuery;
DetailBand1: TQRBand;
qryEmployeeEmpNo:
TIntegerField;
qryEmployeeLastName:
TStringField;
qryEmployeeFirstName:
TStringField;
qryEmployeePhoneExt:
TStringField;
qryEmployeeHireDate:
TDateTimeField;
qryEmployeeSalary: TFloatField;
QRExprMemo1:
TQRExprMemo;
PageHeaderBand1: TQRBand;
QRSysData1: TQRSysData;
<> procedure
DetailBand1BeforePrint(Sender:
TQRCustomBand;
var PrintBand: Boolean);
<> procedure
QuickRep1BeforePrint(Sender:
TCustomQuickRep;
var PrintReport: Boolean);
private
{ Private declarations }
FirstDetail: boolean;
public
{ Public declarations }
end;

QRLabel6: TQRLabel;
QRLabel7: TQRLabel;
QRLabel8: TQRLabel;
PageHeaderBand1: TQRBand;
QRSysData1: TQRSysData;
QRSysData1: 1QRSysData; QRSysData2: TQRSysData;
private
{ Private declarations }
public (Public declarations)
{ Public declarations }
end;
7107
for Constitution Transfer Constitution
frmGrouping: TfrmGrouping;
. 1
implementation
{\$R *.dfm}
,
end.
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
6
{
•••••
:: QuickReport 3.0 for Delphi
3.0/4.0/5.0 ::
::
::
:: Simple report for print the
contents of a stringlist ::
<b>::</b>
:: Copyright (c) 1995-1999 QuSoft
AS ::
:: All Rights Reserved
::
::
•••
:: web: http://www.qusoft.com fax:
+47 22 41 74 91 ::
····· }
unit history;
interface

+47 22 41 74 91 ::
•••••
<b>::::::</b> }
unit grouping;
unit grouping,
• 4 6
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
Db, DBTables, ExtCtrls, QuickRpt,
Orctrls;
type
TfrmGrouping = class(TForm)
QuickRep1: TQuickRep;
MasterQry: TQuery;
MasterQrycompany:
TStringField;
MasterQryorderno: TFloatField;
MasterQrycustno: TFloatField;
MasterQryItemsTotal:
TCurrencyField;
DetailBand1: TQRBand;
QRDBText2: TQRDBText;
QRGroup1: TQRGroup;
QRDBText4: TQRDBText;
DataSource1: TDataSource;
DetailQry: TQuery;
DetailQryOrderNo: TFloatField;
DetailQryQty: TIntegerField;
DetailQryItemNo: TFloatField;
QRSubDetail1: TQRSubDetail;
QRDBText1: TQRDBText;
QRDBText5: TQRDBText;
QRBand1: TQRBand;
QRExpr1: TQRExpr;
QRBand2: TQRBand;
QRExpr2: TQRExpr;
SummaryBand1: TQRBand;
QRExpr3: TQRExpr;
ColumnHeaderBand1: TQRBand;
QRLabel1: TQRLabel;
QRLabel2: TQRLabel;
QRLabel4: TQRLabel;
QRLabel5: TQRLabel;

+47 22 41 74 91 ::
•••••
::::::::}
unit image;
interface
Interrace
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
Qrctrls, QuickRpt, Db, DBTables,
ExtCtrls;
4
type Transfer December 1
TfrmImageRpt = class(TForm)
QuickRep1: TQuickRep;
tbBio: TTable;
DetailBand1: TQRBand;
tbBioSpeciesNo: TFloatField;
tbBioCategory: TStringField;
tbBioCommon Name:
TStringField;
tbBioSpeciesName: TStringField;
tbBioLengthcm: TFloatField;
tbBioLength_In: TFloatField;
tbBioNotes: TMemoField;
tbBioGraphic: TGraphicField;
QRDBText1: TQRDBText;
QRDBImage1: TQRDBImage;
ColumnHeaderBand1: TQRBand;
QRLabel1: TQRLabel;
QRLabel2: TQRLabel;
private
{ Private declarations }
public
{ Public declarations }
end;
var
frmImageRpt: TfrmImageRpt;
implementation
{\$R *.dfm}
(*** ******)
end.
CHU.

****	
uses	davia Magaaga Caallala
	dows, Messages, SysUtils,
	es, Graphics, Controls, Forms,
Dialo	<u> </u>
Qui	ckRpt, Qrctrls, ExtCtrls;
4xxm o	
type	- History — alogg(TForms)
	mHistory = class(TForm)
	ickRep1: TQuickRep;
	tailBand1: TQRBand;
	RMemo1: TQRMemo;
	geHeaderBand1: TQRBand;
	RSysData1: TQRSysData;
priv	
,	rivate declarations }
pub	
,	ublic declarations }
end;	
var	
frm]	History: TfrmHistory;
imple	ementation
{\$R *	.dfm}
end.	
	\\\\\\\\\\\\\\\
7	
{	
::::::	••••••
	•••••
:: Q	uickReport 3.0 for Delphi
3.0/4.	0/5.0 ::
::	
::	
:: Si	mple report with graphics
::	
::	
::	
:: C	opyright (c) 1995-1999 QuSoft
AS	::
:: A	ll Rights Reserved
::	5
::	
::	
:: w	eb: http://www.qusoft.com fax:

0 1 50
qryOrders: TQuery;
QRDBText3: TQRDBText;
QRDBText4: TQRDBText;
dsOrders: TDataSource;
qryItems: TQuery;
QRSubDetailItems:
TQRSubDetail;
QRDBText5: TQRDBText;
QRDBText2: TQRDBText;
QRDBText6: TQRDBText;
QRDBText7: TQRDBText;
QRDBText9: TQRDBText;
QRGroupCust: TQRGroup;
QRLabel1: TQRLabel;
QRLabel2: TQRLabel;
QRShapeGray: TQRShape;
QRLabel3: TQRLabel;
QRLabel4: TQRLabel;
QRLabel5: TQRLabel;
QRLabels: TQRLabel;
QRDBOrderNo: TQRDBText;
QRDBSalesDate: TQRDBText;
GroupFooterBand1: TQRBand;
<> procedure
QRSubDetailOrdersBeforePrint(Sen
der: TQRCustomBand;
var PrintBand: Boolean);
<> procedure
QRSubDetailItemsBeforePrint(Send
er: TQRCustomBand;
var PrintBand: Boolean);
<> procedure
QRGroupCustBeforePrint(Sender:
TQRCustomBand;
var PrintBand: Boolean);
<> procedure
DetailBand1BeforePrint(Sender:
TQRCustomBand;
var PrintBand: Boolean);
<> procedure
QRSubDetailItemsAfterPrint(Sende
r: TQRCustomBand;
BandPrinted: Boolean);
<> procedure
PageHeaderBand1BeforePrint(Send
er: TQRCustomBand;
var PrintBand: Boolean);
vai i i iii Daiiu. Duulkali),
nrivate
private { Private declarations }

111111111111111111111111111111111111111
8
{
••••••
••••••
:: QuickReport 3.0 for Delphi
3.0/4.0/5.0 ::
::
::
:: Master/Detail report with some
extra code ::
::
::
:: Copyright (c) 1995-1999 QuSoft
AS ::
:: All Rights Reserved
::
::
::
:: web: http://www.qusoft.com fax:
+47 22 41 74 91 ::
::::::}
unit mdrpt;
unit murpt,
interface
interrace
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
QuickRpt, Qrctrls, Db, DBTables,
ExtCtrls;
type
TfrmMasterDetail = class(TForm)
QuickRep1: TQuickRep;
DetailBand1: TQRBand;
dsCustomer: TDataSource;
qryCustomer: TQuery;
PageHeaderBand1: TQRBand;
QRDBText1: TQRDBText;
QRSysData1: TQRSysData;
QRSysData1: TQRSysData;
QRSubDetailOrders:
TORSubDetail;
I VINJUDULAII;

end;	public
	{ Public declarations }
<> procedure	end;
TfrmMasterDetail.QRSubDetailOrd	
ersBeforePrint(	var
Sender: TQRCustomBand; var	frmMasterDetail:
PrintBand: Boolean);	TfrmMasterDetail;
begin	
// We are print the order	implementation
information on the first item	
subdetail	{\$R *.dfm}
// record. If there are no item	
records, then we print the subdetail	<> procedure
	TfrmMasterDetail.DetailBand1Befo
// Enable the order fields on the	rePrint(Sender: TQRCustomBand;
item subdetail band. After they are	var PrintBand: Boolean);
// printed once, they will be	begin
disabled until the next order/item set	// If there are no subdetails, then we
QRDBOrderNo.Enabled := true;	print this band. Otherwise we
QRDBSalesDate.Enabled :=	// let the group header print the
QRDBOrderNo.Enabled;	controls from this band. Group
	bands
// Only allow this band to print if	// can be reprinted on page breaks,
there are no subdetails	which detail and subdetails can't do.
qryItems.First;	qryOrders.First;
PrintBand := qryItems.EOF;	PrintBand := qryOrders.EOF;
end;	
,	// If it's our turn to print and the
<> procedure	group band has our controls, then
TfrmMasterDetail.QRSubDetailIte	// we take them back.
msBeforePrint(	if PrintBand and
Sender: TQRCustomBand; var	(Sender.ControlCount = 0) then
PrintBand: Boolean);	with QRGroupCust do
begin	while ControlCount > 0 do
// toggle the item background so	Controls[0].Parent := Sender;
// like the greenbox pener we all	end;
// like the greenbar paper we all know and love.	
	<> procedure
with QRShapeGray.Brush do if Color = \$00F0F0F0 then	TfrmMasterDetail.QRGroupCustBe
Color := \$00E0E0E0	forePrint(Sender:
	TQRCustomBand;
else	var PrintBand: Boolean);
Color := \$00F0F0F0;	begin
end;	// We grab the detail band fields
z > nwo ooduwo	right from under it.
<pre>&lt;&gt; procedure TfrmMastarDetail OPSubDataillta</pre>	if Sender.ControlCount = 0 then
TfrmMasterDetail.QRSubDetailIte msAfterPrint(	with DetailBand1 do
Sender: TQRCustomBand;	While ControlCount > 0 do
SCHUEL. LONGUSHUHDAHU;	Controls[0].Parent := Sender:

unit mdrpt;	BandPrinted: Boolean);
	begin
interface	// After we print it once, we disable
	the controls
uses	QRDBOrderNo.Enabled := false;
Windows, Messages, SysUtils,	QRDBSalesDate.Enabled :=
Classes, Graphics, Controls, Forms,	QRDBOrderNo.Enabled;
Dialogs,	end;
QuickRpt, Qrctrls, Db, DBTables,	
ExtCtrls;	<> procedure
	TfrmMasterDetail.PageHeaderBand
type	1BeforePrint(
TfrmMasterDetail = class(TForm)	Sender: TQRCustomBand; var
QuickRep1: TQuickRep;	PrintBand: Boolean);
DetailBand1: TQRBand;	begin
dsCustomer: TDataSource;	// Re-enable the order fields on the
qryCustomer: TQuery;	item subdetail so that the order
PageHeaderBand1: TQRBand;	// information will be repeated after
QRDBText1: TQRDBText;	a page break
QRSysData1: TQRSysData;	QRDBOrderNo.Enabled := true;
QRSysData2: TQRSysData;	QRDBSalesDate.Enabled :=
QRSubDetailOrders:	QRDBOrderNo.Enabled;
TQRSubDetail;	end;
qryOrders: TQuery;	
QRDBText3: TQRDBText;	end.
QRDBText4: TQRDBText;	
dsOrders: TDataSource;	
qryItems: TQuery;	{
QRSubDetailItems:	
TQRSubDetail;	Owiel Demont 2 0 few Delmhi
QRDBText5: TQRDBText;	:: QuickReport 3.0 for Delphi
QRDBText2: TQRDBText;	3.0/4.0/5.0 ::
QRDBText6: TQRDBText;	
QRDBText7: TQRDBText;	:: Master/Detail report with some
QRDBText9: TQRDBText;	extra code ::
QRGroupCust: TQRGroup;	::
QRLabel1: TQRLabel;	
QRLabel2: TQRLabel;	:: Copyright (c) 1995-1999 QuSoft
QRShapeGray: TQRShape;	AS ::
QRLabel3: TQRLabel;	:: All Rights Reserved
QRLabel4: TQRLabel;	::
QRLabel5: TQRLabel;	::
QRLabel6: TQRLabel;	::
QRDBOrderNo: TQRDBText;	:: web: http://www.qusoft.com fax:
QRDBSalesDate: TQRDBText;	+47 22 41 74 91 ::
GroupFooterBand1: TQRBand;	
<> procedure	
QRSubDetailOrdersBeforePrint(Sen	••••••

PrintBand := qryOrders.EOF;	der: TQRCustomBand;
	var PrintBand: Boolean);
// If it's our turn to print and the	<> procedure
group band has our controls, then	QRSubDetailItemsBeforePrint(Send
// we take them back.	er: TQRCustomBand;
if PrintBand and	var PrintBand: Boolean);
(Sender.ControlCount = 0) then	<> procedure
` '	<u>-</u>
with QRGroupCust do	QRGroupCustBeforePrint(Sender:
while ControlCount > 0 do	TQRCustomBand;
Controls[0].Parent := Sender;	var PrintBand: Boolean);
end;	<> procedure
	DetailBand1BeforePrint(Sender:
<> procedure	TQRCustomBand;
TfrmMasterDetail.QRGroupCustBe	var PrintBand: Boolean);
forePrint(Sender:	<> procedure
TQRCustomBand;	QRSubDetailItemsAfterPrint(Sende
var PrintBand: Boolean);	r: TQRCustomBand;
,	BandPrinted: Boolean);
begin	,,
// We grab the detail band fields	<> procedure
right from under it.	PageHeaderBand1BeforePrint(Send
if Sender.ControlCount = 0 then	er: TQRCustomBand;
with DetailBand1 do	var PrintBand: Boolean);
While ControlCount > 0 do	private
Controls[0].Parent := Sender;	{ Private declarations }
	1.11
end:	public
end;	-
	{ Public declarations }
<> procedure	•
<> procedure TfrmMasterDetail.QRSubDetailOrd	{ Public declarations } end;
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint(	{ Public declarations } end; var
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var	{ Public declarations } end; var frmMasterDetail:
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var PrintBand: Boolean);	{ Public declarations } end; var
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var PrintBand: Boolean); begin	{ Public declarations } end;  var frmMasterDetail: TfrmMasterDetail;
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var PrintBand: Boolean); begin // We are print the order	{ Public declarations } end; var frmMasterDetail:
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var PrintBand: Boolean); begin	{ Public declarations } end;  var frmMasterDetail: TfrmMasterDetail;
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var PrintBand: Boolean); begin // We are print the order	{ Public declarations } end;  var frmMasterDetail: TfrmMasterDetail;
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var PrintBand: Boolean); begin // We are print the order information on the first item	{ Public declarations } end;  var frmMasterDetail: TfrmMasterDetail;  implementation
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var PrintBand: Boolean); begin // We are print the order information on the first item subdetail	{ Public declarations } end;  var frmMasterDetail: TfrmMasterDetail;  implementation
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var PrintBand: Boolean); begin // We are print the order information on the first item subdetail // record. If there are no item	{ Public declarations } end;  var frmMasterDetail: TfrmMasterDetail;  implementation  {\$R *.dfm}  <> procedure
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var PrintBand: Boolean); begin  // We are print the order information on the first item subdetail  // record. If there are no item records, then we print the subdetail	{ Public declarations } end;  var frmMasterDetail: TfrmMasterDetail;  implementation  {\$R *.dfm}  <> procedure TfrmMasterDetail.DetailBand1Befo
<pre>&lt;&gt; procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint(    Sender: TQRCustomBand; var PrintBand: Boolean); begin    // We are print the order information on the first item subdetail    // record. If there are no item records, then we print the subdetail  // Enable the order fields on the</pre>	{Public declarations } end;  var frmMasterDetail: TfrmMasterDetail;  implementation  {\$R *.dfm}  <> procedure TfrmMasterDetail.DetailBand1Befo rePrint(Sender: TQRCustomBand;
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var PrintBand: Boolean); begin  // We are print the order information on the first item subdetail  // record. If there are no item records, then we print the subdetail  // Enable the order fields on the item subdetail band. After they are	{ Public declarations } end;  var frmMasterDetail: TfrmMasterDetail;  implementation  {\$R *.dfm}  <> procedure TfrmMasterDetail.DetailBand1Befo rePrint(Sender: TQRCustomBand; var PrintBand: Boolean);
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var PrintBand: Boolean); begin  // We are print the order information on the first item subdetail  // record. If there are no item records, then we print the subdetail  // Enable the order fields on the item subdetail band. After they are  // printed once, they will be	{Public declarations } end;  var frmMasterDetail: TfrmMasterDetail;  implementation  {\$R *.dfm}  <> procedure TfrmMasterDetail.DetailBand1Befo rePrint(Sender: TQRCustomBand; var PrintBand: Boolean); begin
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var PrintBand: Boolean); begin  // We are print the order information on the first item subdetail  // record. If there are no item records, then we print the subdetail  // Enable the order fields on the item subdetail band. After they are  // printed once, they will be disabled until the next order/item set	{Public declarations } end;  var frmMasterDetail: TfrmMasterDetail;  implementation  {\$R *.dfm}  <> procedure TfrmMasterDetail.DetailBand1Befo rePrint(Sender: TQRCustomBand; var PrintBand: Boolean); begin  // If there are no subdetails, then we
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var PrintBand: Boolean); begin  // We are print the order information on the first item subdetail  // record. If there are no item records, then we print the subdetail  // Enable the order fields on the item subdetail band. After they are  // printed once, they will be disabled until the next order/item set QRDBOrderNo.Enabled := true;	{ Public declarations } end;  var frmMasterDetail: TfrmMasterDetail;  implementation  {\$R *.dfm}  <> procedure TfrmMasterDetail.DetailBand1Befo rePrint(Sender: TQRCustomBand; var PrintBand: Boolean); begin // If there are no subdetails, then we print this band. Otherwise we
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var PrintBand: Boolean); begin  // We are print the order information on the first item subdetail  // record. If there are no item records, then we print the subdetail  // Enable the order fields on the item subdetail band. After they are  // printed once, they will be disabled until the next order/item set QRDBOrderNo.Enabled := true; QRDBSalesDate.Enabled :=	{Public declarations } end;  var frmMasterDetail: TfrmMasterDetail;  implementation  {\$R *.dfm}  <> procedure TfrmMasterDetail.DetailBand1Befo rePrint(Sender: TQRCustomBand; var PrintBand: Boolean); begin  // If there are no subdetails, then we print this band. Otherwise we  // let the group header print the
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var PrintBand: Boolean); begin  // We are print the order information on the first item subdetail  // record. If there are no item records, then we print the subdetail  // Enable the order fields on the item subdetail band. After they are  // printed once, they will be disabled until the next order/item set QRDBOrderNo.Enabled := true;	{Public declarations } end;  var frmMasterDetail: TfrmMasterDetail;  implementation  {\$R *.dfm}  <> procedure TfrmMasterDetail.DetailBand1Befo rePrint(Sender: TQRCustomBand; var PrintBand: Boolean); begin  // If there are no subdetails, then we print this band. Otherwise we  // let the group header print the controls from this band. Group
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var PrintBand: Boolean); begin  // We are print the order information on the first item subdetail  // record. If there are no item records, then we print the subdetail  // Enable the order fields on the item subdetail band. After they are  // printed once, they will be disabled until the next order/item set QRDBOrderNo.Enabled := true; QRDBSalesDate.Enabled :=	{Public declarations } end;  var frmMasterDetail: TfrmMasterDetail;  implementation  {\$R *.dfm}  <> procedure TfrmMasterDetail.DetailBand1Befo rePrint(Sender: TQRCustomBand; var PrintBand: Boolean); begin  // If there are no subdetails, then we print this band. Otherwise we  // let the group header print the
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var PrintBand: Boolean); begin  // We are print the order information on the first item subdetail  // record. If there are no item records, then we print the subdetail  // Enable the order fields on the item subdetail band. After they are  // printed once, they will be disabled until the next order/item set QRDBOrderNo.Enabled := true; QRDBSalesDate.Enabled :=	{Public declarations } end;  var frmMasterDetail: TfrmMasterDetail;  implementation  {\$R *.dfm}  <> procedure TfrmMasterDetail.DetailBand1Befo rePrint(Sender: TQRCustomBand; var PrintBand: Boolean); begin  // If there are no subdetails, then we print this band. Otherwise we  // let the group header print the controls from this band. Group
<> procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var PrintBand: Boolean); begin  // We are print the order information on the first item subdetail  // record. If there are no item records, then we print the subdetail  // Enable the order fields on the item subdetail band. After they are  // printed once, they will be disabled until the next order/item set QRDBOrderNo.Enabled := true; QRDBSalesDate.Enabled;  QRDBOrderNo.Enabled;	Public declarations   end;   var   frmMasterDetail:   TfrmMasterDetail;   implementation   {\$R *.dfm}     <> procedure   TfrmMasterDetail.DetailBand1Befo   rePrint(Sender: TQRCustomBand;   var PrintBand: Boolean);   begin   // If there are no subdetails, then we print this band. Otherwise we   // let the group header print the controls from this band. Group   bands   // can be reprinted on page breaks,
<pre>&lt;&gt; procedure TfrmMasterDetail.QRSubDetailOrd ersBeforePrint( Sender: TQRCustomBand; var PrintBand: Boolean); begin  // We are print the order information on the first item subdetail  // record. If there are no item records, then we print the subdetail  // Enable the order fields on the item subdetail band. After they are  // printed once, they will be disabled until the next order/item set QRDBOrderNo.Enabled := true; QRDBSalesDate.Enabled;  // Only allow this band to print if</pre>	Public declarations   end;   var   frmMasterDetail:   TfrmMasterDetail;   implementation   {\$R *.dfm}     <> procedure   TfrmMasterDetail.DetailBand1Befo   rePrint(Sender: TQRCustomBand;   var PrintBand: Boolean);   begin   // If there are no subdetails, then we print this band. Otherwise we   // let the group header print the controls from this band. Group   bands

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	PrintBand := qryItems.EOF;
9	end;
	,
{	<> procedure
	TfrmMasterDetail.QRSubDetailIte
•••••	msBeforePrint(
:: QuickReport 3.0 for Delphi	Sender: TQRCustomBand; var
3.0/4.0/5.0 ::	PrintBand: Boolean);
::	begin
::	// toggle the item background so
:: Demo report that is populated by	that we can have alternating colors
the OnNeedData event ::	// like the greenbar paper we all
::	know and love.
::	with QRShapeGray.Brush do
:: Copyright (c) 1995-1999 QuSoft	if Color = \$00F0F0F0 then
AS ::	Color := \$00E0E0E0
:: All Rights Reserved	else
::	Color := \$00F0F0F0;
::	end;
::	
:: web: http://www.qusoft.com fax:	<> procedure
+47 22 41 74 91 ::	TfrmMasterDetail.QRSubDetailIte
	msAfterPrint(
	Sender: TQRCustomBand;
:::::::::::}	BandPrinted: Boolean);
unit needdata;	begin
	// After we print it once, we disable
interface	the controls
	QRDBOrderNo.Enabled := false;
uses	QRDBSalesDate.Enabled :=
Windows, Messages, SysUtils,	QRDBOrderNo.Enabled;
Classes, Graphics, Controls, Forms,	end;
Dialogs,	_
ExtCtrls, QuickRpt, Qrctrls;	<> procedure
4	TfrmMasterDetail.PageHeaderBand
type Tree No. 10. (TE	1BeforePrint(
TfrmNeedData = class(TForm)	Sender: TQRCustomBand; var
QuickRep1: TQuickRep;	PrintBand: Boolean);
DetailBand1: TQRBand;	begin
QRLabel1: TQRLabel;	// Re-enable the order fields on the
TitleBand1: TQRBand;	item subdetail so that the order
QRSysData1: TQRSysData;	// information will be repeated after
<> procedure FormCroats(Sondor: TObject):	a page break
FormCreate(Sender: TObject); <> procedure	QRDBOrderNo.Enabled := true; QRDBSalesDate.Enabled :=
QuickRep1BeforePrint(Sender:	QRDBOrderNo.Enabled;
TCustomQuickRep;	end;
var PrintReport: Boolean);	ciiu,
<> procedure	and
procedure	end.

// If MoreData is true, then
QuickReport will print
// another detail band. When you
set it to false,
// the report is done.
" the report is done.
MoreData := (CurrentIndex <
SomeList.Count);
Some Distrebuilty,
if MoreData then
begin
QRLabel1.Caption :=
SomeList[CurrentIndex];
Some Dist[Currentinuex],
// Here's how to set the progress
bar
QuickRep1.QRPrinter.Progress :=
(Longint(CurrentIndex) * 100) div
SomeList.Count;
end
else
QuickRep1.QRPrinter.Progress :=
100;
100;
In a (Current Indox).
Inc(CurrentIndex);
end;
and a
end.
10
10
{
0 1 D 4206 D 1 1
:: QuickReport 3.0 for Delphi
3.0/4.0/5.0/6.0 ::
::
::
:: Example reports project
::
<b>:</b>
:: Copyright (c) 1995-1999 QuSoft
AS ::
:: All Rights Reserved
::
::
::
:: web: http://www.qusoft.com fax:

QuickRep1NeedData(Sender:
TObject; var MoreData: Boolean);
private
_
{ Private declarations }
SomeList: TStringlist;
CurrentIndex: integer;
public
{ Public declarations }
end;
var
frmNeedData: TfrmNeedData;
,
implementation
Implementation
(AD 4 16 )
{\$R *.dfm}
<> procedure
TfrmNeedData.FormCreate(Sender:
TObject);
var
i: integer;
begin
SomeList := TStringlist.Create;
for i := 0 to 500 do
SomeList.Add('Line ' +
· ·
IntToStr(i));
end;
<> procedure
TfrmNeedData.QuickRep1BeforePri
nt(Sender: TCustomQuickRep;
var PrintReport: Boolean);
begin
// You must reset your data in the
BeforePrint event
// or when you print from the
preview, the report will
// start with the last value(s)
CurrentIndex := 0;
end;
<> procedure
TfrmNeedData.QuickRep1NeedData
=
(Sender: TObject;
var MoreData: Boolean);
begin

Label5: TLabel;
rbNeedData: TRadioButton;
rbFormLetter: TRadioButton;
btnExport: TButton;
SaveDialog1: TSaveDialog;
rbAbout: TRadioButton;
rbGrouping: TRadioButton;
Image2: TImage;
<> procedure
CreateList(Sender: TObject);
<> procedure
QRCompositeReport1AddReports(S
ender: TObject);
<> procedure
btnCRClick(Sender: TObject);
<> procedure
rbCreateListClick(Sender:
TObject);
<> procedure
rbMasterDetailClick(Sender:
TObject);
<> procedure
FormCreate(Sender: TObject);
<> procedure
FormActivate(Sender: TObject);
<> procedure
btnPreviewClick(Sender: TObject);
<> procedure
rbExprMemoClick(Sender:
TObject);
<> procedure
rbImageClick(Sender: TObject);
<> procedure
rbBasicMDClick(Sender: TObject);
<> procedure
btnFiltersClick(Sender: TObject);
<> procedure
rbCompositeClick(Sender:
TObject);
<pre>&lt;&gt; procedure rhNoodDataClick(Sandar: TObject);</pre>
rbNeedDataClick(Sender: TObject);
<> procedure
rbFormLetterClick(Sender:
TObject);
<pre>&lt;&gt; procedure btnEvnortClick(Sondar: TObject);</pre>
btnExportClick(Sender: TObject);
<> procedure
rbAboutClick(Sender: TObject);
<> procedure

+47 22 41 74 91 ::
:::::::::::}
unit mdmain;
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
ExtCtrls, StdCtrls, quickrpt, Db,
DBTables, printers, grextra,
QRExport,
qrprntr;
const
Composite_Description = 'The
composite control can be used to link
several'+
' reports together as a
single report.';
type
TfrmQR3Demo = class(TForm)
QRTextFilter1: TQRTextFilter;
QRCSVFilter1: TQRCSVFilter;
QRHTMLFilter1:
TQRHTMLFilter;
QRCompositeReport1:
TQRCompositeReport;
Label1: TLabel;
VersionLbl: TLabel;
Label3: TLabel;
GroupBox1: TGroupBox;
rbCreateList: TRadioButton;
btnPreview: TButton;
btnPrint: TButton;
rbMasterDetail: TRadioButton;
Description: TMemo;
•
OpenDialog1: TOpenDialog;
cbPreview: TComboBox;
Label4: TLabel;
rbExprMemo: TRadioButton;
rbImage: TRadioButton;
rbBasicMD: TRadioButton;

MyTable: TTable;	rbGroupingClick(Sender: TObject);
	private
nIdx: integer;	-
begin	{ Private declarations }
{ Create a table on the fly, this	FReport: TCustomQuickRep;
example uses a table from the demo	CreateListReport: TQuickRep;
database }	<> procedure
MyTable := TTable.Create(self);	SetReport(Value :
	TCustomQuickRep);
{ create the list of fields to output	public
from the table }	{ Public declarations }
SomeFields := TStringList.Create;	property Report :
with MyTable do	TCustomQuickRep read FReport
begin	write SetReport;
DatabaseName := 'DBDEMOS';	end;
TableName := 'COUNTRY.DB';	- Chu,
ReadOnly := True;	var
	frmQR3Demo: TfrmQR3Demo;
Active := True;	mmyksbemo: mmyksbemo;
// 7	• • •
// For this example, we will pull	implementation
the field names from the table	
// If you wanted to only use some	uses mdrpt, exprmemo, image,
of the fields, you would edit	basicmd, needdata, frmltr,
// this list.	history, grouping;
for nIdx := 0 to FieldCount - 1 do	
	{\$R *.dfm}
SomeFields.Add(Fields[nIdx].FieldN	
ame);	<> procedure
end;	TfrmQR3Demo.SetReport(Value :
	TCustomQuickRep);
// If you didn't create the report,	begin
you must set the report object to nil	FReport := Value;
// before calling QRCreateList	if Value $\Leftrightarrow$ nil then
	if Value =
areport := nil;	TCustomQuickRep(QRCompositeR
	eport1) then
{ Build the report }	Description.Lines.Text :=
	Composite_Description
// If you change the displaywidth, it	else
will be reflecte in the created	
// report	Description.Lines.Assign(Report.Des
•	cription);
with MyTable.Fields[1] do	end;
DisplayWidth := DisplayWidth	Ź
div 2;	<> procedure
uiv 29	TfrmQR3Demo.CreateList(Sender:
// areata the report	TObject);
// create the report	¥ //
QRCreateList(aReport, nil,	a Donort a TCustom Quick Dono
MyTable, 'Country Listing',	aReport: TCustomQuickRep;
SomeFields);	SomeFields: TStringList;

end;	
end	// Make the column header's font
else if sender = btnPrint then	use bold attribute
	use bold attilibute
areport.print;	aDanant Danda Calumu HaadauDd
	aReport.Bands.ColumnHeaderBand.
{ all done, free the objects }	Font.Style := [fsBold];
aReport.Free;	
MyTable.Free;	// Now adjust the spacing of the
SomeFields.Free;	fields. There isn't any reason to
end;	// do this, this is just to show how to
	access the controls on the
<> procedure	// report.
TfrmQR3Demo.QRCompositeRepor	
t1AddReports(Sender: TObject);	for nIdx := 0 to
begin	aReport.Bands.ColumnHeaderBand.
// The OnAddReports event is	ControlCount -1 do
called by the CompositeReport	if
// to add the reports to list of	aReport.Bands.ColumnHeaderBand.
-	Controls[nIdx] is TQRPrintable
reports	then
with	with
QRCompositeReport1.Reports do	TQRPrintable(aReport.Bands.Colu
begin	` -
	mnHeaderBand.Controls[nIdx]) do
Add(frmMasterDetail.QuickRep1);	Left := Left - (5 * nIdx);
Add(frmBasicMD.QuickRep1);	C 11 0 4
Add(frmImageRpt.QuickRep1);	for $nIdx := 0$ to
end;	aReport.Bands.DetailBand.ControlC
end;	ount -1 do
	if
<> procedure	aReport.Bands.DetailBand.Controls[
TfrmQR3Demo.btnCRClick(Sender	nIdx] is TQRPrintable then
: TObject);	with
begin	TQRPrintable(aReport.Bands.Detail
QRCompositeReport1.Preview;	Band.Controls[nIdx]) do
end;	Left := Left - $(5 * nIdx)$ ;
Citu,	
< > nwooodure	
<> procedure Thur OD 2 Dome ab Cheeta List Click (S	{ You can change the report objects
TfrmQR3Demo.rbCreateListClick(S	before calling the report }
ender: TObject);	// areport.page.orientation :=
begin	poLandscape;
Report := CreateListReport;	{preview or print the report}
end;	{preview or print the report}
	if and an - htm Durania th
<> procedure	if sender = btnPreview then
TfrmQR3Demo.rbMasterDetailClic	begin
k(Sender: TObject);	case cbPreview.ItemIndex of
begin	0: areport.preview;
Report :=	1: areport.previewModal;
frmMasterDetail.QuickRep1;	2: areport.previewModeless;
¿	

if report = CreateListReport then CreateList(Sender) else if report = TCustomQuickRep(QRCompositeR eport1) then begin if sender = btnPreview then QRCompositeReport1.preview else QRCompositeReport1.print;	end;  <> pr TfrmQR3D : TObject); var i: intege
else if report = TCustomQuickRep(QRCompositeR eport1) then begin if sender = btnPreview then QRCompositeReport1.preview else	TfrmQR3D : TObject); var i: intege
TCustomQuickRep(QRCompositeR eport1) then begin if sender = btnPreview then QRCompositeReport1.preview else	TfrmQR3D : TObject); var i: intege
eport1) then begin if sender = btnPreview then QRCompositeReport1.preview else	: TObject); var i: intege
begin if sender = btnPreview then QRCompositeReport1.preview else	var i: intege
if sender = btnPreview then QRCompositeReport1.preview else	
QRCompositeReport1.preview else	begin
else	
	// Get the c
ORCompositeReport1.print:	version nun
I	// the qrpr
end	i := pos(' ',
else	if $i \ge 0$ the
begin	VersionL
if sender = btnPreview then	copy(cQRN
begin	// Create tl
case cbPreview.ItemIndex of	the QRCrea
0: report.preview;	CreateList
1: report.previewModal;	TQuickRep
2: report.previewModeless;	CreateList
end;	:= 'Example
end	QRCreateL
else if sender = btnPrint then	cbPreview
report.print;	end;
end;	
end;	<> pr
end;	TfrmQR3D
enu;	er: TObject
	begin
<> procedure	if Descript
TfrmQR3Demo.rbExprMemoClick(	begin
Sender: TObject);	rbCreate
begin	rbCreate
Report :=	end;
frmExprmemo.QuickRep1;	end;
end;	enu,
	<> pr
<> procedure	TfrmQR3D
TfrmQR3Demo.rbImageClick(Send	nder: TObj
er: TObject);	begin
begin	// This cod
Report :=	
frmImageRpt.QuickRep1;	than what y
end;	// typically
	Most of this
<> procedure	// is to han
TfrmQR3Demo.rbBasicMDClick(Se	various type
nder: TObject);	// reports.
begin	
Report :=	if report <
frmBasicMD.QuickRep1;	begin

•
end;
<> procedure
TfrmQR3Demo.FormCreate(Sender
: TObject);
var i: integer;
begin
// Get the current QuickReport
version number from
// the qrprntr unit
i := pos(' ', cQRName);
if i >=0 then
VersionLbl.Caption := 'Version' +
<u>=</u>
copy(cQRName, i, 99);
// Create the report object used by
the QRCreateList() function
CreateListReport :=
TQuickRep.Create(self);
CreateListReport.Description.Text
:= 'Example of how to call the
QRCreateList function';
cbPreview.ItemIndex := 0;
end;
<> procedure
TfrmQR3Demo.FormActivate(Send
er: TObject);
begin
if Description.Lines.Count = 0 then
begin
<u> </u>
rbCreateListClick(Self);
rbCreateList.Checked := True;
end;
end;
<> procedure
TfrmQR3Demo.btnPreviewClick(Se
nder: TObject);
begin
// This code is more complicated
than what you would
// typically need to run a report.
Most of this code
// is to handle the selection of the
various types of
// reports.
10 ( 0 11.1
if report <> nil then
begin

if Execute then
begin
frmFormLetter.QuickRep1.ExportT
oFilter(TQRCommaSeparatedFilter.
Create(FileName));
{
Other filters:
HTML: TQRHTMLDocumentFilter
ASCII: TQRAsciiExportFilter
CSV: TQRCommaSeparatedFilter
In Professional Version:
RTF: TQRRTFExportFilter
WMF: TQRWMFExportFilter
Excel: TQRXLSFilter
}
end;
end;
btnExport.Enabled := True;
end;
<> procedure
TfrmQR3Demo.rbAboutClick(Send
er: TObject);
begin
Report := frmHistory.QuickRep1;
end;
<> procedure
TfrmQR3Demo.rbGroupingClick(Se
nder: TObject);
begin
Report :=
frmGrouping.QuickRep1;
end;
end.

end;
<> procedure
TfrmQR3Demo.btnFiltersClick(Sen
der: TObject);
begin
MessageDlg('When an export filter
control is dropped on a form, it''s
added to all of the
previews',mtInformation, [mbok],
0);
end;
<pre>&lt;&gt; procedure TfrmOD2Domo nh Composito Click(S)</pre>
TfrmQR3Demo.rbCompositeClick(S
ender: TObject);
begin Persont
Report := TCystomOviolyBoy(OBCommositeB
TCustomQuickRep(QRCompositeR
eport1);
end;
Z Nuncoedune
<> procedure TfrmQR3Demo.rbNeedDataClick(S
ender: TObject);
• /:
begin Panant :-
Report := frmNeedDate OwiekBen1:
frmNeedData.QuickRep1; end;
enu,
<> procedure
TfrmQR3Demo.rbFormLetterClick(
Sender: TObject);
begin
Report :=
frmFormLetter.QuickRep1;
end;
······
// The following code show how to
explicitly call an export
// filter without going through the
preview
<> procedure
TfrmQR3Demo.btnExportClick(Sen
der: TObject);
begin
btnExport.Enabled := False;
with SaveDialog1 do
begin

' reports together as a	
single report.';	الملف التنفيذي لبرنامج
type	(Qr3)
TfrmQR3Demo = class(TForm)	(Q13)
QRTextFilter1: TQRTextFilter;	إعداد ا
QRCSVFilter1: TQRCSVFilter;	1 111 11 41
QRHTMLFilter1:	علاء الدين اللباد
TQRHTMLFilter;	واته ۷۳۵۷۵۲۹۰
QRCompositeReport1:	
TQRCompositeReport;	علاء الدين اللباد علاء الدين اللباد
Label1: TLabel;	
VersionLbl: TLabel;	procedures 110
Label3: TLabel;	:: QuickReport 3.0 for Delphi
GroupBox1: TGroupBox;	3.0/4.0/5.0/6.0 ::
rbCreateList: TRadioButton;	::
btnPreview: TButton;	::
btnPrint: TButton;	:: Example reports project
rbMasterDetail: TRadioButton;	::
Description: TMemo;	::
OpenDialog1: TOpenDialog;	::
cbPreview: TComboBox;	:: Copyright (c) 1995-1999 QuSoft
Label4: TLabel;	AS ::
rbExprMemo: TRadioButton;	:: All Rights Reserved
rbImage: TRadioButton;	::
rbBasicMD: TRadioButton;	::
btnFilters: TButton;	::
rbComposite: TRadioButton;	:: web: http://www.qusoft.com fax:
Label5: TLabel;	+47 22 41 74 91 ::
rbNeedData: TRadioButton;	
rbFormLetter: TRadioButton;	
btnExport: TButton;	·············}
SaveDialog1: TSaveDialog;	unit mdmain;
rbAbout: TRadioButton;	
rbGrouping: TRadioButton;	interface
Image2: TImage;	
<> procedure	uses
CreateList(Sender: TObject);	Windows, Messages, SysUtils,
<> procedure	Classes, Graphics, Controls, Forms,
QRCompositeReport1AddReports(S	Dialogs,
ender: TObject);	ExtCtrls, StdCtrls, quickrpt, Db,
<> procedure	DBTables, printers, qrextra,
btnCRClick(Sender: TObject);	QRExport,
<> procedure	qrprntr;
rbCreateListClick(Sender:	
TObject);	const
<> procedure	Composite_Description = 'The
rbMasterDetailClick(Sender:	composite control can be used to link
TObject);	several'+

hasiamd panddata fumltu	Z Nuccedune
basicmd, needdata, frmltr,	<> procedure Form Create (Sondary TObject):
history, grouping;	FormCreate(Sender: TObject);
(OD 1, 16 )	<> procedure
{\$R *.dfm}	FormActivate(Sender: TObject);
	<> procedure
<> procedure	btnPreviewClick(Sender: TObject);
TfrmQR3Demo.SetReport(Value:	<> procedure
TCustomQuickRep);	rbExprMemoClick(Sender:
begin	TObject);
FReport := Value;	<> procedure
if Value <> nil then	rbImageClick(Sender: TObject);
if Value =	<> procedure
TCustomQuickRep(QRCompositeR	rbBasicMDClick(Sender: TObject);
eport1) then	<> procedure
Description.Lines.Text :=	btnFiltersClick(Sender: TObject);
Composite Description	<> procedure
else	rbCompositeClick(Sender:
CISC	TObject);
Description Lines Assign/Deport Des	<> procedure
Description.Lines.Assign(Report.Des	rbNeedDataClick(Sender: TObject);
cription);	<> procedure
end;	-
	rbFormLetterClick(Sender:
<> procedure	TObject);
TfrmQR3Demo.CreateList(Sender:	<> procedure
TObject);	btnExportClick(Sender: TObject);
var	<> procedure
aReport : TCustomQuickRep;	rbAboutClick(Sender: TObject);
SomeFields: TStringList;	<> procedure
MyTable: TTable;	rbGroupingClick(Sender: TObject);
nIdx: integer;	private
begin	{ Private declarations }
{ Create a table on the fly, this	FReport : TCustomQuickRep;
example uses a table from the demo	CreateListReport : TQuickRep;
database }	<> procedure
MyTable := TTable.Create(self);	SetReport(Value :
injudic. Habic.ereate(seii),	TCustomQuickRep);
{ create the list of fields to output	public
_ ·	{ Public declarations }
from the table }	property Report :
SomeFields := TStringList.Create;	TCustomQuickRep read FReport
with MyTable do	write SetReport;
begin	end;
DatabaseName := 'DBDEMOS';	ciiu,
TableName := 'COUNTRY.DB';	
ReadOnly := True;	var
Active := True;	frmQR3Demo: TfrmQR3Demo;
// For this example, we will pull	implementation
the field names from the table	
// If you wanted to only use some	uses mdrpt, exprmemo, image,
II John miled to only use some	

TQRPrintable(aReport.Bands.Colu	of the fields, you would edit
mnHeaderBand.Controls[nIdx]) do	// this list.
Left := Left - (5 * nIdx);	for nIdx := 0 to FieldCount - 1 do
for nIdx := 0 to	SomeFields.Add(Fields[nIdx].FieldN
aReport.Bands.DetailBand.ControlC	ame);
ount -1 do	end;
if	
aReport.Bands.DetailBand.Controls[	// If you didn't create the report,
nIdx] is TQRPrintable then	you must set the report object to nil
with	// before calling QRCreateList
TQRPrintable(aReport.Bands.Detail	
Band.Controls[nIdx]) do	areport := nil;
<b>Left</b> := <b>Left</b> - (5 * n <b>Id</b> x);	_
	{ Build the report }
{ You can change the report objects	// If you change the displaywidth, it
before calling the report }	will be reflecte in the created
// areport.page.orientation :=	// report
poLandscape;	
{preview or print the report}	with MyTable.Fields[1] do
	DisplayWidth := DisplayWidth
if sender = btnPreview then	div 2;
begin	
case cbPreview.ItemIndex of	// create the report
0: areport.preview;	QRCreateList(aReport, nil,
1: areport.previewModal;	MyTable, 'Country Listing',
2: areport.previewModeless;	SomeFields);
end;	
end	// Make the column header's font
else if sender = btnPrint then	use bold attribute
areport.print;	
	aReport.Bands.ColumnHeaderBand.
{ all done, free the objects }	Font.Style := [fsBold];
aReport.Free;	// NJ 1. 4.41 . 6.43
MyTable.Free;	// Now adjust the spacing of the
SomeFields.Free;	fields. There isn't any reason to
end;	// do this, this is just to show how to
	access the controls on the
<> procedure	// report.
TfrmQR3Demo.QRCompositeRepor	for nIdx := 0 to
t1AddReports(Sender: TObject);	
begin // The On Add Deports event is	aReport.Bands.ColumnHeaderBand. ControlCount -1 do
// The OnAddReports event is	if
called by the CompositeReport	aReport.Bands.ColumnHeaderBand.
// to add the reports to list of	Controls[nIdx] is TQRPrintable
reports	then
with OPCompositePenert1 Penerts do	with
QRCompositeReport1.Reports do	771111

cbPreview.ItemIndex := 0;
end;
<> procedure
TfrmQR3Demo.FormActivate(Send
er: TObject);
begin
if Description.Lines.Count = 0 then
begin
rbCreateListClick(Self);
rbCreateList.Checked := True;
end;
end;
<> procedure
TfrmQR3Demo.btnPreviewClick(Se
nder: TObject);
begin
// This code is more complicated
than what you would
// typically need to run a report.  Most of this code
// is to handle the selection of the
various types of
// reports.
// Teports.
if report <> nil then
begin
if report = CreateListReport then
CreateList(Sender)
else if report =
TCustomQuickRep(QRCompositeR
eport1) then
begin
if sender = btnPreview then
QRCompositeReport1.preview
else
QRCompositeReport1.print;
end
else
begin
if sender = btnPreview then
begin
case cbPreview.ItemIndex of
0: report.preview;
1: report.previewModal;
2: report.previewModeless;
end;

begin
Add(frmMasterDetail.QuickRep1);
Add(frmBasicMD.QuickRep1);
Add(frmImageRpt.QuickRep1);
end;
end;
enu;
<> procedure
TfrmQR3Demo.btnCRClick(Sender
: TObject);
begin
QRCompositeReport1.Preview;
end;
<> procedure
TfrmQR3Demo.rbCreateListClick(S
ender: TObject);
begin
Report := CreateListReport;
end;
,
<> procedure
TfrmQR3Demo.rbMasterDetailClic
k(Sender: TObject);
begin
Report :=
frmMasterDetail.QuickRep1;
end;
Ciru,
<> procedure
TfrmQR3Demo.FormCreate(Sender
: TObject);
var i: integer;
begin
// Get the current QuickReport
version number from
// the qrprntr unit
i := pos(' ', cQRName); if i >=0 then
VersionLbl.Caption := 'Version' +
copy(cQRName, i, 99);
// Create the report object used by
the QRCreateList() function
CreateListReport :=
TQuickRep.Create(self);
CreateListReport.Description.Text
:= 'Example of how to call the
QRCreateList function';

end;	end
chu,	else if sender = btnPrint then
/ > nmaaaduma	
<> procedure TfrmQR3Demo.rbNeedDataClick(S	report.print;
`	end;
ender: TObject);	end;
begin	end;
Report :=	
frmNeedData.QuickRep1;	<> procedure
end;	TfrmQR3Demo.rbExprMemoClick(
	Sender: TObject);
<> procedure	begin
TfrmQR3Demo.rbFormLetterClick(	Report :=
Sender: TObject);	frmExprmemo.QuickRep1;
begin	end;
Report :=	,
frmFormLetter.QuickRep1;	<> procedure
end;	TfrmQR3Demo.rbImageClick(Send
	er: TObject);
// The following code show how to	begin
explicitly call an export	Report :=
// filter without going through the	frmImageRpt.QuickRep1;
preview	end;
<> procedure	cnu,
<u> </u>	
TfrmQR3Demo.btnExportClick(Sen	<> procedure
der: TObject);	TfrmQR3Demo.rbBasicMDClick(Se
begin	nder: TObject);
btnExport.Enabled := False;	begin
with SaveDialog1 do	Report :=
begin	frmBasicMD.QuickRep1;
if Execute then	end;
begin	
	<> procedure
frmFormLetter.QuickRep1.ExportT	TfrmQR3Demo.btnFiltersClick(Sen
oFilter(TQRCommaSeparatedFilter.	der: TObject);
Create(FileName));	begin
{	MessageDlg('When an export filter
Other filters:	control is dropped on a form, it''s
HTML: TQRHTMLDocumentFilter	added to all of the
ASCII: TQRAsciiExportFilter	previews',mtInformation, [mbok],
CSV: TQRCommaSeparatedFilter	0);
	end;
In Professional Version:	
RTF: TQRRTFExportFilter	<> procedure
WMF: TQRWMFExportFilter	TfrmQR3Demo.rbCompositeClick(S
Excel: TQRXLSFilter	ender: TObject);
PACCI. I VINALISTIICI	begin
) and	Report :=
end;	TCustomQuickRep(QRCompositeR
end;	
btnExport.Enabled := True;	eport1);

Classes, Graphics, Controls, Forms,
Dialogs, Db, DBTables, ExtCtrls, QuickRpt,
Oretrls;
Q100135
type
TfrmBasicMD = class(TForm)
QuickRep1: TQuickRep;
dsCustomer: TDataSource;
tbCustomer: TTable;
DetailBand1: TQRBand;
QRSubDetail1: TQRSubDetail;
QRDBText1: TQRDBText;
QRDBText2: TQRDBText;
qryOrders: TQuery;
QRGroup1: TQRGroup;
QRDBText3: TQRDBText;
ColumnHeaderBand1: TQRBand; QRLabel1: TQRLabel;
QRLabel: TQRLabel; QRLabel2: TQRLabel;
QRLabel3: TQRLabel;
QRDBText4: TQRDBText;
QRLabel4: TQRLabel;
QRBand1: TQRBand;
QRExpr1: TQRExpr;
QRLabel5: TQRLabel;
private
{ Private declarations }
public
{ Public declarations }
end;
var
frmBasicMD: TfrmBasicMD;
implementation
(CD * dfm)
{\$R *.dfm}
end.
Citu.
\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
3
{
••••••

end;
<> procedure
TfrmQR3Demo.rbAboutClick(Send
er: TObject);
begin
Report := frmHistory.QuickRep1;
end;
<> procedure
TfrmQR3Demo.rbGroupingClick(Se
nder: TObject);
begin
Report :=
frmGrouping.QuickRep1;
end;
,
end.
\\\\\\\\\\\
2
{
••••••
:: QuickReport 3.0 for Delphi
3.0/4.0/5.0 ::
::
:: Basic Master/Detail report with
group bands ::
••
:: Copyright (c) 1995-1999 QuSoft
AS ::
:: All Rights Reserved
An Rights Reserved
••
••
:: web: http://www.qusoft.com fax:
+47 22 41 74 91 ::
17/4471/71 ··
unit basicmd;
unit vasicinu,
interface
шетасе
uses
Windows, Messages, SysUtils,

TStringField;	
tbCustomerLastInvoiceDate:	
TDateTimeField;	
QRExprMemo1:	
TQRExprMemo;	
private	
{ Private declarations }	
public	
{ Public declarations }	
end;	
var	
frmExprmemo: TfrmExprmemo;	
implementation	
-	
{\$R *.dfm}	
(422 1822)	
end.	
\\\\\\\\\	
4	
{	
••••••	
***************************************	
:: QuickReport 3.0 for Delphi	
3.0/4.0/5.0 ::	
<b>::</b>	
::	
:: Simple report for doing a form	
letter ::	
::	
:: Copyright (c) 1995-1999 QuSoft	
AS ::	
:: All Rights Reserved	
Ali Kigilis Keserveu	
••	
::	
•	
:: web: http://www.qusoft.com fax:	
+47 22 41 74 91 ::	
<b></b> }	
unit frmltr;	
,	
interface	
1111111111111	
NCOS	
uses — — — — — — — — — — — — — — — — — —	
Windows, Messages, SysUtils,	

:: QuickReport 3.0 for Delphi
3.0/4.0/5.0 ::
::
::
:: Demonstration on how to use the
TQRExprMemo control ::
::
::
:: Copyright (c) 1995-1999 QuSoft AS ::
:: All Rights Reserved
::
::
::
:: web: http://www.qusoft.com fax: +47 22 41 74 91 ::
***************************************
······ }
unit exprmemo;
•
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
QuickRpt, Qrctrls, Db, DBTables,
ExtCtrls;
type
TfrmExprmemo = class(TForm)
QuickRep1: TQuickRep;
tbCustomer: TTable;
DetailBand1: TQRBand;
tbCustomerCustNo: TFloatField;
tbCustomerCompany:
TStringField;
tbCustomerAddr1: TStringField;
tbCustomerAddr2: TStringField;
tbCustomerCity: TStringField;
tbCustomerState: TStringField;
tbCustomerZip: TStringField;
tbCustomerCountry:
TStringField;
tbCustomerPhone: TStringField;
tbCustomerFAX: TStringField;
tbCustomerTaxRate: TFloatField;
tbCustomerContact:

Print(Sender: TQRCustomBand;	Classes, Graphics, Controls, Forms,
var PrintBand: Boolean);	Dialogs,
begin	Db, DBTables, ExtCtrls, QuickRpt,
if FirstDetail then	Qrctrls;
FirstDetail := False	
else	type
QuickRep1.NewPage;	TfrmFormLetter = class(TForm)
end;	QuickRep1: TQuickRep;
	qryEmployee: TQuery;
<> procedure	DetailBand1: TQRBand;
TfrmFormLetter.QuickRep1Before	qryEmployeeEmpNo:
Print(Sender: TCustomQuickRep;	TIntegerField;
var PrintReport: Boolean);	qryEmployeeLastName:
begin	TStringField;
FirstDetail := True;	qryEmployeeFirstName:
end;	TStringField;
	qryEmployeePhoneExt:
end.	TStringField;
Cita.	qryEmployeeHireDate:
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	TDateTimeField;
5	qryEmployeeSalary: TFloatField;
3	QRExprMemo1:
<u> </u>	TQRExprMemo;
}	PageHeaderBand1: TQRBand;
	QRSysData1: TQRSysData;
:: QuickReport 3.0 for Delphi	<> procedure
3.0/4.0/5.0 ::	DetailBand1BeforePrint(Sender:
::	TQRCustomBand;
	var PrintBand: Boolean);
:: Simple report for grouping data	<> procedure
::	QuickRep1BeforePrint(Sender:
••	TCustomQuickRep;
	var PrintReport: Boolean);
:: Copyright (c) 1995-1999 QuSoft	private
AS ::	{ Private declarations }
:: All Rights Reserved	FirstDetail: boolean;
::	public
::	{ Public declarations }
::	end;
:: web: http://www.qusoft.com fax:	/
+47 22 41 74 91 ::	var
	frmFormLetter: TfrmFormLetter;
······································	implementation
unit grouping;	
9-7	{\$R *.dfm}
interface	(*** ******)
	<> procedure
uses	TfrmFormLetter.DetailBand1Before
usos	THE OF INDICATED AND THE COURT OF

{ Public declarations }
end;
var
frmGrouping: TfrmGrouping;
rmGrouping: TrmGrouping;
• •
implementation
{\$R *.dfm}
end.
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
6
U
(
{
••••••
***************************************
:: QuickReport 3.0 for Delphi
3.0/4.0/5.0 ::
••
::
:: Simple report for print the
contents of a stringlist ::
::
:: Copyright (c) 1995-1999 QuSoft
AS ::
:: All Rights Reserved
::
::
<b>::</b>
:: web: http://www.qusoft.com fax:
+47 22 41 74 91 ::
***************************************
mit history
unit history;
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
QuickRpt, Qrctrls, ExtCtrls;
Quickithi, Qienis, Exicuis,
type
TfrmHistory = class(TForm)

W:1 M CII4:1-
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
Db, DBTables, ExtCtrls, QuickRpt,
Qrctrls;
type
TfrmGrouping = class(TForm)
QuickRep1: TQuickRep;
MasterQry: TQuery;
MasterQrycompany:
TStringField;
MasterQryorderno: TFloatField;
MasterQrycustno: TFloatField;
MasterQryItemsTotal:
TCurrencyField;
DetailBand1: TQRBand;
QRDBText2: TQRDBText;
QRGroup1: TQRGroup;
QRDBText4: TQRDBText;
DataSource1: TDataSource;
DetailQry: TQuery;
DetailQryOrderNo: TFloatField;
DetailQryQty: TIntegerField;
DetailQryItemNo: TFloatField;
QRSubDetail1: TQRSubDetail;
QRDBText1: TQRDBText;
QRDBText5: TQRDBText;
QRBand1: TQRBand;
QRExpr1: TQRExpr;
QRBand2: TQRBand;
QRExpr2: TQRExpr;
SummaryBand1: TQRBand;
QRExpr3: TQRExpr;
ColumnHeaderBand1: TQRBand;
QRLabel1: TQRLabel;
QRLabel2: TQRLabel;
QRLabel4: TQRLabel;
QRLabel5: TQRLabel;
QRLabel6: TQRLabel;
QRLabel7: TQRLabel;
QRLabel8: TQRLabel;
PageHeaderBand1: TQRBand;
QRSysData1: TQRSysData;
QRSysData2: TQRSysData;
private
{ Private declarations }
public

Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
Qrctrls, QuickRpt, Db, DBTables,
ExtCtrls;
type
TfrmImageRpt = class(TForm)
QuickRep1: TQuickRep;
tbBio: TTable;
DetailBand1: TQRBand;
tbBioSpeciesNo: TFloatField;
tbBioCategory: TStringField;
tbBioCommon Name:
TStringField;
tbBioSpeciesName: TStringField;
tbBioLengthcm: TFloatField;
tbBioLength_In: TFloatField;
tbBioNotes: TMemoField;
tbBioGraphic: TGraphicField;
QRDBText1: TQRDBText;
QRDBImage1: TQRDBImage;
ColumnHeaderBand1: TQRBand;
QRLabel1: TQRLabel;
QRLabel2: TQRLabel;
private
{ Private declarations }
public
{ Public declarations }
end;
var
frmImageRpt: TfrmImageRpt;
implementation
-
{\$R *.dfm}
(4-1-1411)
end.
Cirus
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
8
0
(
{
0.110
:: QuickReport 3.0 for Delphi
3.0/4.0/5.0 ::

QuickRep1: TQuickRep;	
DetailBand1: TQRBand;	
QRMemo1: TQRMemo;	
PageHeaderBand1: TQRBand;	
QRSysData1: TQRSysData;	
private	
{ Private declarations }	
public	
{ Public declarations }	
end;	
cnu,	
VOW	
fun Historya Tfun Historya	
frmHistory: TfrmHistory;	
implementation	
implementation	
(CD + Jf)	
{\$R *.dfm}	
and a	
end.	
7	
{	
•••••	
•••••	
:: QuickReport 3.0 for Delphi	
3.0/4.0/5.0 ::	
::	
::	
:: Simple report with graphics	
::	
::	
::	
:: Copyright (c) 1995-1999 QuSoft	
AS ::	
:: All Rights Reserved	
::	
::	
::	
:: web: http://www.qusoft.com fax:	
+47 22 41 74 91 ::	
::::::}	
unit image;	
interface	
uses	

<u></u>	
QRDBText6: TQRDBText;	
QRDBText7: TQRDBText;	
QRDBText9: TQRDBText;	
QRGroupCust: TQRGroup;	
QRLabel1: TQRLabel;	
QRLabel2: TQRLabel;	
QRShapeGray: TQRShape;	
QRLabel3: TQRLabel;	
QRLabel4: TQRLabel;	
QRLabel5: TQRLabel;	
QRLabel6: TQRLabel;	
QRDBOrderNo: TQRDBText;	
QRDBSalesDate: TQRDBText;	
GroupFooterBand1: TQRBand;	
<> procedure	
QRSubDetailOrdersBeforePrint(Sen	
der: TQRCustomBand;	
var PrintBand: Boolean);	
<> procedure	
QRSubDetailItemsBeforePrint(Send	
er: TQRCustomBand;	
var PrintBand: Boolean);	
<> procedure	
QRGroupCustBeforePrint(Sender:	
TQRCustomBand;	
var PrintBand: Boolean);	
<> procedure	
DetailBand1BeforePrint(Sender:	
TQRCustomBand;	
var PrintBand: Boolean);	
<> procedure	
QRSubDetailItemsAfterPrint(Sende	
r: TQRCustomBand;	
BandPrinted: Boolean);	
<> procedure	
PageHeaderBand1BeforePrint(Send	
er: TQRCustomBand;	
var PrintBand: Boolean);	
private	
{ Private declarations }	
public	
{ Public declarations }	
end;	
,	
var	
frmMasterDetail:	
TfrmMasterDetail;	
implementation	
impicination and in	

:: .:
:: Master/Detail report with some
_
extra code ::
::
:: G
:: Copyright (c) 1995-1999 QuSoft
AS ::
:: All Rights Reserved
::
::
::
:: web: http://www.qusoft.com fax:
+47 22 41 74 91 ::
::::::::::}
unit mdrpt;
univ mui pu
interface
Interrace
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
QuickRpt, Qrctrls, Db, DBTables,
ExtCtrls;
type
TfrmMasterDetail = class(TForm)
QuickRep1: TQuickRep;
DetailBand1: TQRBand;
dsCustomer: TDataSource;
qryCustomer: TQuery;
PageHeaderBand1: TQRBand;
QRDBText1: TQRDBText;
QRSysData1: TQRSysData;
QRSysData2: TQRSysData;
QRSubDetailOrders:
TQRSubDetail;
qryOrders: TQuery;
QRDBText3: TQRDBText;
QRDBText4: TQRDBText;
dsOrders: TDataSource;
qryItems: TQuery;
QRSubDetailItems:
TQRSubDetail;
QRDBText5: TQRDBText;
·
QRDBText2: TQRDBText;

information on the first item	
subdetail	{\$R *.dfm}
// record. If there are no item	
records, then we print the subdetail	<> procedure
	TfrmMasterDetail.DetailBand1Befo
// Enable the order fields on the	rePrint(Sender: TQRCustomBand;
item subdetail band. After they are	var PrintBand: Boolean);
// printed once, they will be	begin
disabled until the next order/item set	// If there are no subdetails, then we
QRDBOrderNo.Enabled := true;	print this band. Otherwise we
QRDBSalesDate.Enabled :=	// let the group header print the
QRDBOrderNo.Enabled;	controls from this band. Group
	bands
// Only allow this band to print if	// can be reprinted on page breaks,
there are no subdetails	which detail and subdetails can't do.
qryItems.First;	qryOrders.First;
PrintBand := qryItems.EOF;	PrintBand := qryOrders.EOF;
end;	
	// If it's our turn to print and the
<> procedure	group band has our controls, then
TfrmMasterDetail.QRSubDetailIte	// we take them back.
msBeforePrint(	if PrintBand and
Sender: TQRCustomBand; var	(Sender.ControlCount = 0) then
PrintBand: Boolean);	with QRGroupCust do
begin	while ControlCount > 0 do
// toggle the item background so	Controls[0].Parent := Sender;
that we can have alternating colors	end;
// like the greenbar paper we all	enu;
know and love.	2 Summer dum.
	<pre>&lt;&gt; procedure Tform Manday Data il ODG varia ContData </pre>
with QRShapeGray.Brush do	TfrmMasterDetail.QRGroupCustBe
if Color = \$00F0F0F0 then	forePrint(Sender:
Color := \$00E0E0E0	TQRCustomBand;
else	var PrintBand: Boolean);
Color := \$00F0F0F0;	begin
end;	// We grab the detail band fields
	right from under it.
<> procedure	if Sender.ControlCount = 0 then
TfrmMasterDetail.QRSubDetailIte	with DetailBand1 do
msAfterPrint(	While ControlCount > 0 do
Sender: TQRCustomBand;	Controls[0].Parent := Sender;
BandPrinted: Boolean);	end;
begin	
// After we print it once, we disable	<> procedure
the controls	TfrmMasterDetail.QRSubDetailOrd
QRDBOrderNo.Enabled := false;	ersBeforePrint(
QRDBSalesDate.Enabled :=	Sender: TQRCustomBand; var
QRDBOrderNo.Enabled;	PrintBand: Boolean);
end;	begin
	// We are print the order
	4

ExtCtrls;
type
TfrmMasterDetail = class(TForm)
QuickRep1: TQuickRep;
DetailBand1: TQRBand;
dsCustomer: TDataSource;
qryCustomer: TQuery;
PageHeaderBand1: TQRBand;
QRDBText1: TQRDBText;
QRSysData1: TQRSysData;
QRSysData2: TQRSysData;
QRSubDetailOrders:
TQRSubDetail;
qryOrders: TQuery;
QRDBText3: TQRDBText;
QRDBText4: TQRDBText;
dsOrders: TDataSource;
qryItems: TQuery;
QRSubDetailItems:
TQRSubDetail;
QRDBText5: TQRDBText;
QRDBText2: TQRDBText;
QRDBText6: TQRDBText;
QRDBText7: TQRDBText;
QRDBText9: TQRDBText;
QRGroupCust: TQRGroup;
QRLabel1: TQRLabel;
QRLabel2: TQRLabel;
QRShapeGray: TQRShape;
QRLabel3: TQRLabel;
QRLabel4: TQRLabel;
QRLabel5: TQRLabel;
QRLabel6: TQRLabel;
QRDBOrderNo: TQRDBText;
QRDBSalesDate: TQRDBText;
GroupFooterBand1: TQRBand;
<> procedure
QRSubDetailOrdersBeforePrint(Sen
der: TQRCustomBand;
var PrintBand: Boolean);
<> procedure
QRSubDetailItemsBeforePrint(Send
er: TQRCustomBand;
var PrintBand: Boolean);
<> procedure
QRGroupCustBeforePrint(Sender:
TQRCustomBand;

∠ > nuonaduun
<> procedure
TfrmMasterDetail.PageHeaderBand
1BeforePrint(
Sender: TQRCustomBand; var
PrintBand: Boolean);
begin
// Re-enable the order fields on the
item subdetail so that the order
// information will be repeated after
a page break
QRDBOrderNo.Enabled := true;
QRDBSalesDate.Enabled :=
QRDBOrderNo.Enabled;
end;
end.
CHU.
111111111111111111111111111111111111111
{
•••••
•••••
:: QuickReport 3.0 for Delphi
3.0/4.0/5.0 ::
••
::
:: Master/Detail report with some
extra code ::
::
::
:: Copyright (c) 1995-1999 QuSoft
AS ::
:: All Rights Reserved
An Rights Reserved
••
::
::
:: web: http://www.qusoft.com fax:
+47 22 41 74 91 ::
::::::::::}
unit mdrpt;
աու ուս բւ,
6
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
QuickRpt, Orctrls, Db, DBTables,
- vaichithe victib, DD, DD I auich

Controls[0].Parent := Sender;	var PrintBand: Boolean);
end;	<> procedure
chu,	DetailBand1BeforePrint(Sender:
<> procedure	TQRCustomBand;
TfrmMasterDetail.QRGroupCustBe	var PrintBand: Boolean);
forePrint(Sender:	<> procedure
TQRCustomBand;	QRSubDetailItemsAfterPrint(Sende
var PrintBand: Boolean);	r: TQRCustomBand;
begin	BandPrinted: Boolean);
// We grab the detail band fields	<> procedure
right from under it.	PageHeaderBand1BeforePrint(Send
if Sender.ControlCount = 0 then	er: TQRCustomBand;
with DetailBand1 do	var PrintBand: Boolean);
While ControlCount > 0 do	private
Controls[0].Parent := Sender;	{ Private declarations }
end;	public
Ciiu,	{ Public declarations }
/ > nroandura	end;
<> procedure TfrmMasterDetail.QRSubDetailOrd	onu,
ersBeforePrint(	var
Sender: TQRCustomBand; var	frmMasterDetail:
PrintBand: Boolean);	TfrmMasterDetail;
begin	Tirimviaster Detail,
// We are print the order	implementation
information on the first item	Implementation
subdetail	{\$R *.dfm}
// record. If there are no item	ξΦK .uim;
records, then we print the subdetail	<> procedure
records, then we print the subdetain	TfrmMasterDetail.DetailBand1Befo
// Enable the order fields on the	rePrint(Sender: TQRCustomBand;
item subdetail band. After they are	var PrintBand: Boolean);
// printed once, they will be	begin
disabled until the next order/item set	// If there are no subdetails, then we
QRDBOrderNo.Enabled := true;	print this band. Otherwise we
QRDBSalesDate.Enabled :=	// let the group header print the
QRDBOrderNo.Enabled;	controls from this band. Group
	bands
// Only allow this band to print if	// can be reprinted on page breaks,
there are no subdetails	which detail and subdetails can't do.
qryItems.First;	qryOrders.First;
PrintBand := qryItems.EOF;	PrintBand := qryOrders.EOF;
end;	Timebana , qiyotacis.Doi ,
w	// If it's our turn to print and the
<> procedure	group band has our controls, then
TfrmMasterDetail.QRSubDetailIte	// we take them back.
msBeforePrint(	if PrintBand and
Sender: TQRCustomBand; var	(Sender.ControlCount = 0) then
PrintBand: Boolean);	with QRGroupCust do
begin	while ControlCount > 0 do
No. Sim	white Control Occupit > 0 uo

::	// toggle th
:: Demo report that is populated by	that we can
the OnNeedData event ::	// like the g
::	know and lo
<b>::</b>	with QRSh
:: Copyright (c) 1995-1999 QuSoft	if Color =
AS ::	Color :=
:: All Rights Reserved	else
::	Color :=
::	end;
::	322.029
:: web: http://www.qusoft.com fax:	<> pr
+47 22 41 74 91 ::	TfrmMaster
	msAfterPri
	Sender: TO
:::::::::::}	BandPrinte
unit needdata;	begin
	// After we
interface	the controls
merace	QRDBOrd
uses	QRDBSale
Windows, Messages, SysUtils,	QRDBOrde
Classes, Graphics, Controls, Forms,	end;
Dialogs,	enu,
ExtCtrls, QuickRpt, Qrctrls;	<> pr
Externs, Quickiept, Quents,	TfrmMaster
type	1BeforePrin
TfrmNeedData = class(TForm)	Sender: TO
QuickRep1: TQuickRep;	PrintBand:
DetailBand1: TQRBand;	begin
	// Re-enabl
QRLabel1: TQRLabel;	
TitleBand1: TQRBand;	item subdet
QRSysData1: TQRSysData;	// informat
<> procedure	a page break
FormCreate(Sender: TObject);	QRDBOrd
<> procedure	QRDBSale
QuickRep1BeforePrint(Sender:	QRDBOrde
TCustomQuickRep;	end;
var PrintReport: Boolean);	1
<> procedure	end.
QuickRep1NeedData(Sender:	\\\\\\\
TObject; var MoreData: Boolean);	9
private	
{ Private declarations }	{
SomeList: TStringlist;	•••••
CurrentIndex: integer;	
public	:: QuickRe
{ Public declarations }	3.0/4.0/5.0
end;	<b>::</b>

e item background so have alternating colors reenbar paper we all apeGray.Brush do \$00F0F0F0 then **\$00E0E0E0 \$00F0F0F0;** ocedure Detail.QRSubDetailIte ORCustomBand; d: Boolean); print it once, we disable erNo.Enabled := false; sDate.Enabled := rNo.Enabled; ocedure Detail.PageHeaderBand QRCustomBand; var Boolean); e the order fields on the ail so that the order ion will be repeated after erNo.Enabled := true; sDate.Enabled := rNo.Enabled; //////// port 3.0 for Delphi ::

begin
QRLabel1.Caption :=
SomeList[CurrentIndex];
// Here's how to set the progress
bar
QuickRep1.QRPrinter.Progress :=
(Longint(CurrentIndex) * 100) div
SomeList.Count;
end
else
QuickRep1.QRPrinter.Progress :=
- 1
100;
Inc(CurrentIndex);
end;
end.
10
{
•••••
:: QuickReport 3.0 for Delphi
3.0/4.0/5.0/6.0 ::
::
::
:: Example reports project
::
:: Copyright (c) 1995-1999 QuSoft
AS ::
:: All Rights Reserved
:: 
:: web: http://www.qusoft.com fax:
+47 22 41 74 91 ::
······································
unit mdmain;
interface
uses
Windows, Messages, SysUtils,

var
frmNeedData: TfrmNeedData;
implementation
mprementation
{\$R *.dfm}
(wit dilli)
<> procedure
TfrmNeedData.FormCreate(Sender:
`
TObject);
· · · · · · · · · · · · · · · · · · ·
i: integer;
begin
SomeList := TStringlist.Create;
for i := 0 to 500 do
SomeList.Add('Line ' +
IntToStr(i));
end;
<> procedure
TfrmNeedData.QuickRep1BeforePri
nt(Sender: TCustomQuickRep;
var PrintReport: Boolean);
begin
// You must reset your data in the
BeforePrint event
// or when you print from the
preview, the report will
// start with the last value(s)
CurrentIndex := 0;
end;
<> procedure
TfrmNeedData.QuickRep1NeedData
(Sender: TObject;
var MoreData: Boolean);
begin
// If MoreData is true, then
QuickReport will print
// another detail band. When you
set it to false,
// the report is done.
MoreData := (CurrentIndex <
SomeList.Count);
Some Distriction 199
if MoreData then
n Mivi CData tiitii

Came procedure QRCompositeReportIAddReports(Sender: TObject);  Came procedure btnCRClick(Sender: TObject);  Came procedure rbCreateListClick(Sender: TObject);  Came procedure rbMasterPetailClick(Sender: TObject);  Came procedure rbMasterPetailClick(Sender: TObject);  Came procedure rbmActivate(Sender: TObject);  Came procedure rbExprMemoClick(Sender: TObject);  Came procedure rbExprMemoClick(Sender: TObject);  Came procedure rbEmageClick(Sender: TObject);  Came procedure rbImageClick(Sender: TObject);  Came procedure rbCompositeClick(Sender: TObject);  Came procedure rbComposi	Cycatal ist(Sanday TObicat).	Classes Cumbies Controls Forms
QRCompositeReport1AddReports(Sender: TObject);	CreateList(Sender: TObject);	Classes, Graphics, Controls, Forms,
ender: TObject); <pre></pre>	_	8 /
QRExport,   qrprntr;		
btnCRClick(Sender: TObject);		
<pre>came &gt; procedure rbCreateListClick(Sender: TObject);  <pre>came &gt; procedure rbMasterDetailClick(Sender: TObject);  <pre>came &gt; procedure FormCreate(Sender: TObject);  <pre>came &gt; procedure FormActivate(Sender: TObject);  <pre>came &gt; procedure btnPreviewClick(Sender: TObject);  <pre>came &gt; procedure rbExprMemoClick(Sender: TObject);  <pre>came &gt; procedure rbExprMemoClick(Sender: TObject);  <pre>came &gt; procedure rbExprMemoClick(Sender: TObject);  <pre>came &gt; procedure rbEmasicMDClick(Sender: TObject);  <pre>came &gt; procedure rbCompositeClick(Sender: TObject);  <pre>came &gt; procedure rbCompositeClick(Sender: TObject);  <pre>came &gt; procedure rbFormLetterClick(Sender: TObject);  <pre>came &gt; procedure rbAboutClick(Sender: TObject);  <pre>came &gt; procedure rbGroupingClick(Sender: TObject);  <pre>came &gt; procedure rbAboutClick(Sender: TObject);  <pre>ca</pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>	_	•
rbCreateListClick(Sender: TObject):		qrprntr;
TObject); <pre></pre>	-	
<pre>composite control can be used to link several'+  Tobject);  composite control can be used to link several'+  ' reports together as a single report.';  type  TfmQR3Demo = class(TForm) QRCSVFilter1: TQRCSVFilter; QRCSVFilter1: TQRCSVFilter; QRCSVFilter1: TQRCSVFilter; QRCSVFilter1: TQRCSVFilter; QRCOmpositeReport1: TQRCOmpositeReport1: TQRCOMPOSITEREPORT2: TABOBLET TRAGIOBUTON; TbCreateList: TRAGIOBUTON; TbCreateList: TRAGIOBUTON; TbCreateList: TButton; TbCreateList: Tabulton; TbCreateList. Tabulton; TbCr</pre>	· ·	const
rbMasterDetailClick(Sender: TObject);  <> procedure FormCreate(Sender: TObject);  <> procedure thrPreviewClick(Sender: TObject);  <> procedure rbExprMemoClick(Sender: TObject);  <> procedure rbImageClick(Sender: TObject);  <> procedure rbBasicMDClick(Sender: TObject);  <> procedure rbCompositeClick(Sender: TObject);  <> procedure rbTobject);  <> procedure rbCompositeClick(Sender: TObject);  <> procedure rbCompositeClick(Sender: TObject); > procedure rbCompositeClick(Sender: TObject); > procedure rbAboutClick(Sender: TObject); > procedure rbGroupingClick(Sender: TObject); > procedure rbGroupingClick(Sender: TObject); > procedure rbGroupingClick(Sender: TObject); > procedure	• / /	Composite_Description = 'The
TObject);  <	<> procedure	composite control can be used to link
<pre> &lt;&gt; procedure FormCreate(Sender: TObject); &lt;&gt; procedure btnPreviewClick(Sender: TObject); &lt;&gt; procedure rbExprMemoClick(Sender: TObject); &lt;&gt; procedure rbImageClick(Sender: TObject); &lt;&gt; procedure rbBasicMDClick(Sender: TObject); &lt;&gt; procedure rbCompositeClick(Sender: TObject); &lt;&gt; procedure rbCompositeClick(Sender: TObject); &lt;&gt; procedure rbNeedDataClick(Sender: TObject); &lt;&gt; procedure rbNeedDataClick(Sender: TObject); &lt;&gt; procedure rbNeedDataClick(Sender: TObject); &lt;&gt; procedure rbNeedDataClick(Sender: TObject); &lt;&gt; procedure btnExportClick(Sender: TObject); &lt;&gt; procedure btnExportClick(Sender: TObject); &lt;&gt; procedure rbAboutClick(Sender: TObject); &lt;&gt; procedure rbAboutClick(Sender: TObject);&gt; procedure fbFormLetterClick(Sender: TObject);</pre>	rbMasterDetailClick(Sender:	several'+
FormCreate(Sender: TObject);  <> procedure btnPreviewClick(Sender: TObject);  <> procedure rbExprMemoClick(Sender: TObject);  <> procedure rbImageClick(Sender: TObject);  <> procedure rbBasicMDClick(Sender: TObject);  <> procedure rbBasicMDClick(Sender: TObject);  <> procedure rbEsprMemoClick(Sender: TObject);  <> procedure rbImageClick(Sender: TObject);  <> procedure rbFoiltersClick(Sender: TObject);  <> procedure rbNeedDataClick(Sender: TObject);  <> procedure rbFormLetterClick(Sender: TObject);  <> procedure rbFormLetterClick(Sender: TObject);  <> procedure rbAboutClick(Sender: TObject);  <> procedure rbGroupingClick(Sender: TObject);  private {     Private declarations }     FReport : TCustomQuickRep;     CreateListReport : TQuickRep;    > procedure     StReport(Value :     TCustomQuickRep);  <> procedure     StReport(Value :     TCustomQuickRep);	TObject);	' reports together as a
<pre></pre>	<> procedure	single report.';
FormActivate(Sender: TObject);  <> procedure btnPreviewClick(Sender: TObject);  <> procedure rbExprMemoClick(Sender: TObject);  <> procedure rbImageClick(Sender: TObject);  <> procedure rbBasicMDClick(Sender: TObject);  <> procedure rbCompositeClick(Sender: TObject);  <> procedure rbCompositeClick(Sender: TObject);  <> procedure rbCompositeClick(Sender: TObject);  <> procedure rbCompositeClick(Sender: TObject);  <> procedure rbFormLetterClick(Sender: TObject);  <> procedure rbFormLetterClick(Sender: TObject);  <> procedure rbAboutClick(Sender: TObject);  <> procedure rbGroupingClick(Sender: TObject);  <> procedure fbGroupingClick(Sender: TObject); > procedure fPFrivate declarations } FReport: TCustomQuickRep; CreateListReport: TQuickRep;> procedure SetReport(Value: TCustomQuickRep);	FormCreate(Sender: TObject);	
FormActivate(Sender: TObject);  <> procedure btnPreviewClick(Sender: TObject);  <> procedure rbExprMemoClick(Sender: TObject);  <> procedure rbImageClick(Sender: TObject);  <> procedure rbBasicMDClick(Sender: TObject);  <> procedure rbCompositeClick(Sender: TObject);  <> procedure rbCompositeClick(Sender: TObject);  <> procedure rbCompositeClick(Sender: TObject);  <> procedure rbCompositeClick(Sender: TObject);  <> procedure rbFormLetterClick(Sender: TObject);  <> procedure rbFormLetterClick(Sender: TObject);  <> procedure rbAboutClick(Sender: TObject);  <> procedure rbGroupingClick(Sender: TObject);  <> procedure fbGroupingClick(Sender: TObject); > procedure fPFrivate declarations } FReport: TCustomQuickRep; CreateListReport: TQuickRep;> procedure SetReport(Value: TCustomQuickRep);		type
<pre></pre>	-	
btnPreviewClick(Sender: TObject);  <> procedure rbExprMemoClick(Sender: TObject);  <> procedure rbImageClick(Sender: TObject);  <> procedure rbBasicMDClick(Sender: TObject);  <> procedure btnFiltersClick(Sender: TObject);  <> procedure rbCompositeClick(Sender: TObject);  <> procedure rbNeedDataClick(Sender: TObject);  <> procedure rbFormLetterClick(Sender: TObject);  <> procedure rbFormLetterClick(Sender: TObject);  <> procedure rbTobject);  <> procedure rbAboutClick(Sender: TObject);  <> procedure rbAboutClick(Sender: TObject);  <> procedure rbAboutClick(Sender: TObject);  <> procedure rbAboutClick(Sender: TObject); private {		·
<pre> &lt;&gt; procedure rbExprMemoClick(Sender:     TObject);  &lt;&gt; procedure rbImageClick(Sender: TObject);  &lt;&gt; procedure rbBasicMDClick(Sender: TObject);  &lt;&gt; procedure btnFiltersClick(Sender: TObject);  &lt;&gt; procedure rbCompositeClick(Sender: TObject);  &lt;&gt; procedure rbNeedDataClick(Sender: TObject);  &lt;&gt; procedure rbFormLetterClick(Sender: TObject);  &lt;&gt; procedure rbFormLetterClick(Sender: TObject);  &lt;&gt; procedure rbExportClick(Sender: TObject);  &lt;&gt; procedure rbAboutClick(Sender: TObject);  &lt;&gt; procedure rbAboutClick(Sender: TObject);  &lt;&gt; procedure rbAboutClick(Sender: TObject);  &lt;&gt; procedure rbForupingClick(Sender: TObject); &gt; procedure     { Private declarations }     FReport : TCustomQuickRep;     C-reateListReport : TQuickRep;    &gt; procedure     SetReport(Value :     TCustomQuickRep);      TObject); &gt; procedure SetReport(Value :     TCustomQuickRep);      Tobject); </pre>	-	
rbExprMemoClick(Sender: TObject);  <> procedure rbImageClick(Sender: TObject);  <> procedure rbBasicMDClick(Sender: TObject);  <> procedure btnFiltersClick(Sender: TObject);  <> procedure rbCompositeClick(Sender: TObject);  <> procedure rbNeedDataClick(Sender: TObject);  <> procedure rbFormLetterClick(Sender: TObject);  <> procedure btnExportClick(Sender: TObject);  <> procedure btnExportClick(Sender: TObject);  <> procedure rbAboutClick(Sender: TObject);  <> procedure rbGroupingClick(Sender: TObject);  <> procedure {		
TObject);  <> procedure rbImageClick(Sender: TObject); <> procedure rbBasicMDClick(Sender: TObject); <> procedure rbTiltersClick(Sender: TObject); <> procedure rbCompositeClick(Sender: TObject); <> procedure rbNeedDataClick(Sender: TObject); <> procedure rbFormLetterClick(Sender: TObject); <> procedure rbTobject); <> procedure rbTobject); <> procedure rbFormLetterClick(Sender: TObject); <> procedure rbTobject); <> procedure rbAboutClick(Sender: TObject); <> procedure rbAboutClick(Sender: TObject); <> procedure rbGroupingClick(Sender: TObject);> procedure {		
<pre>&lt;&gt; procedure rbImageClick(Sender: TObject); &lt;&gt; procedure rbBasicMDClick(Sender: TObject); &lt;&gt; procedure btnFiltersClick(Sender: TObject); &lt;&gt; procedure rbCompositeClick(Sender: TObject); &lt;&gt; procedure rbNeedDataClick(Sender: TObject); &lt;&gt; procedure rbFormLetterClick(Sender: TObject); &lt;&gt; procedure rbFormLetterClick(Sender: TObject); &lt;&gt; procedure rbAboutClick(Sender: TObject); &lt;&gt; procedure rbAboutClick(Sender: TObject); &lt;&gt; procedure rbGroupingClick(Sender: TObject); &lt;&gt; procedure rbFormLetterClick(Sender: TObject); &lt;&gt; procedure rbAboutClick(Sender: TObject); &lt;&gt; procedure rbGroupingClick(Sender: TObject);&gt; procedure rbGroupingClick(Sender: TObject);&gt; procedure setReport : TCustomQuickRep; CreateListReport : TQuickRep;&gt; procedure SetReport(Value : TCustomQuickRep);&gt; procedure SetReport(Value : TCustomQuickRep);&gt; procedure SetReport(Value : TCustomQuickRep);&gt; procedure&gt; procedure&gt; procedure&gt; procedure</pre>		
rbImageClick(Sender: TObject);  <> procedure rbBasicMDClick(Sender: TObject);  <> procedure btnFiltersClick(Sender: TObject);  <> procedure rbCompositeClick(Sender: TObject);  <> procedure rbNeedDataClick(Sender: TObject);  <> procedure rbFormLetterClick(Sender: TObject);  <> procedure rbFormLetterClick(Sender: TObject);  <> procedure rbAboutClick(Sender: TObject);  <> procedure rbAboutClick(Sender: TObject);  <> procedure rbAboutClick(Sender: TObject);  <> procedure rbForupingClick(Sender: TObject); > procedure rbForupingClick(Sender: TObject); > procedure rbForupingClick(Sender: TObject); > procedure fbroupingClick(Sender: TObject); > procedure setReport: TCustomQuickRep; > procedure SetReport(Value: TCustomQuickRep);	• / /	
<pre>&lt;&gt; procedure rbBasicMDClick(Sender: TObject); &lt;&gt; procedure btnFiltersClick(Sender: TObject); &lt;&gt; procedure rbCompositeClick(Sender: TObject); &lt;&gt; procedure rbNeedDataClick(Sender: TObject); &lt;&gt; procedure rbFormLetterClick(Sender: TObject); &lt;&gt; procedure rbFormLetterClick(Sender: TObject); &lt;&gt; procedure rbtnExportClick(Sender: TObject); &lt;&gt; procedure rbAboutClick(Sender: TObject); &lt;&gt; procedure rbAboutClick(Sender: TObject); &lt;&gt; procedure rbGroupingClick(Sender: TObject); private { Private declarations } FReport : TCustomQuickRep; CreateListReport : TQuickRep; &lt;&gt; procedure SetReport(Value : TCustomQuickRep);</pre> VersionLbl: TLabel; Label3: TLabel; GroupBox1: TGadioButton; btnPreview: TButton; rbMasterDetail: TRadioButton; Description: TMemo; OpenDialog1: TOpenDialog; cbPreview: TComboBox; Label4: TLabel; rbExprMemo: TRadioButton; rbBasicMD: TRadioButton; rbComposite: TRadioButton; btnExport: TButton; SaveDialog1: TSaveDialog; rbAbout: TRadioButton; rbGrouping: TRadioButton; Image2: TImage;  NewsonLibl: TLabel; GroupBox1: TGroupBox; rbCreateList: TRadioButton; btnPrint: TButton; Description: TMemo; OpenDialog1: TOpenDialog; cbPreview: TComboBox; Label4: TLabel; rbExprMemo: TRadioButton; rbComposite: TRadioButton; shall in the print: TButton; rbExprMemo: TRadioButton; rbComposite: TRadioButton; rbFormLetter: TRadioButton; rbComposite: TRadioButton; rbComposite: TRadioButton; rbFormLetter: TRadioButton; rbFormLetter: TRadioButton; rbComposite: TRadioButton; rbFormLetter: TRadioButton; rbFormLetter: TRadioButton; rbComposite: TRadioButton; rbFormLetter: TRadio	-	
rbBasicMDClick(Sender: TObject);  <> procedure btnFiltersClick(Sender: TObject);  <> procedure rbCompositeClick(Sender: TObject);  <> procedure rbNeedDataClick(Sender: TObject);  <> procedure rbFormLetterClick(Sender: TObject);  <> procedure rbTobject);  <> procedure rbTobject);  <> procedure btnExportClick(Sender: TObject);  <> procedure rbAboutClick(Sender: TObject);  <> procedure rbGroupingClick(Sender: TObject); > procedure { Private declarations } FReport: TCustomQuickRep;		· ·
<pre></pre>	-	
btnFiltersClick(Sender: TObject);  <> procedure rbCompositeClick(Sender: TObject);  <> procedure rbNeedDataClick(Sender: TObject);  <> procedure rbFormLetterClick(Sender: TObject);  <> procedure btnExportClick(Sender: TObject);  <> procedure rbAboutClick(Sender: TObject);  <> procedure rbAboutClick(Sender: TObject);  <> procedure rbGroupingClick(Sender: TObject);  private { Private declarations } FReport: TCustomQuickRep;  CreateListReport: TQuickRep;  <> procedure SetReport(Value: TCustomQuickRep);   SINCEPPENT STORAGIOBUTTON; btnPrivate IntradioButton; btnPreview: TButton;  Description: TMemo; OpenDialog1: TOpenDialog; cbPreview: TComboBox; Label4: TLabel; rbExprMemo: TRadioButton; rbImage: TRadioButton; btnFilters: TButton; rbComposite: TRadioButton; btnExport: TButton; saveDialog1: TSaveDialog; rbAbout: TRadioButton; Image2: TImage;	` **	
<pre>c&gt; procedure rbCompositeClick(Sender:     TObject);</pre>	-	
rbCompositeClick(Sender: TObject);  <> procedure rbNeedDataClick(Sender: TObject);  <> procedure rbFormLetterClick(Sender: TObject);  <> procedure btnExportClick(Sender: TObject);  <> procedure rbAboutClick(Sender: TObject);  <> procedure rbAboutClick(Sender: TObject);  <> procedure rbGroupingClick(Sender: TObject);  private { Private declarations } FReport: TCustomQuickRep; CreateListReport: TQuickRep;  <> procedure SetReport(Value: TCustomQuickRep);  btnPrint: TButton; rbMasterDetail: TRadioButton;  Description: TMemo; OpenDialog1: TOpenDialog; cbPreview: TComboBox; Label4: TLabel; rbExprMemo: TRadioButton; rbBasicMD: TRadioButton; btnFilters: TButton; rbComposite: TRadioButton; btnExport: TButton; SaveDialog1: TSaveDialog; rbAbout: TRadioButton; rbGrouping: TRadioButton; Image2: TImage;		· · · · · · · · · · · · · · · · · · ·
TObject); <pre></pre>	=	
<pre>c&gt; procedure rbNeedDataClick(Sender: TObject);</pre>	• `	· ·
rbNeedDataClick(Sender: TObject); <> procedure rbFormLetterClick(Sender: TObject); <> procedure btnExportClick(Sender: TObject); <> procedure rbAboutClick(Sender: TObject); <> procedure rbGroupingClick(Sender: TObject); private { Private declarations } FReport: TCustomQuickRep; CreateListReport: TQuickRep; <> procedure SetReport(Value: TCustomQuickRep);    SetTption: TARMING, OpenDialog1: TOpenDialog; cbPreview: TComboBox; Label4: TLabel; rbExprMemo: TRadioButton; rbBasicMD: TRadioButton; btnFilters: TButton; rbComposite: TRadioButton; rbFormLetter: TRadioButton; SaveDialog1: TSaveDialog; rbAbout: TRadioButton; Image2: TImage;	• / /	· · · · · · · · · · · · · · · · · · ·
<pre>&lt;&gt; procedure rbFormLetterClick(Sender:     TObject);     &lt;&gt; procedure btnExportClick(Sender: TObject);     &lt;&gt; procedure rbAboutClick(Sender: TObject);     &lt;&gt; procedure rbGroupingClick(Sender: TObject);     private     { Private declarations }     FReport: TCustomQuickRep;     CreateListReport: TQuickRep;     &lt;&gt; procedure SetReport(Value:     TCustomQuickRep);</pre> CreateUstomQuickRep);  cbPreview: TComboBox; Label4: TLabel; rbExprMemo: TRadioButton; rbBasicMD: TRadioButton; rbComposite: TRadioButton; rbFormLetter: TRadioButton; saveDialog1: Topenbalog, cbPreview: TComboBox; Label4: TLabel; rbBasicMD: TRadioButton; rbComposite: TRadioButton; rbFormLetter: TRadioButton; SaveDialog1: Topenbalog, cbPreview: TComboBox; Label4: TLabel; rbBasicMD: TRadioButton; rbComposite: TRadioButton; rbFormLetter: TRadioButton; saveDialog1: Topenbalog, rbBasicMD: TRadioButton; rbComposite: TRadioButton; rbFormLetter: TRadioButton; Image2: TImage; rbAbout: TRadioButton; rbFormLetter: TRadioButton; rbFormLetter: TRadioButton; saveDialog1: Topenbalog, rbAboutclick(Sender: TObject); rbBasicMD: TRadioButton; rbComposite: TRadioButton; rbFormLetter: TRadioButton; saveDialog1: Topenbalog, rbAboutclick(Sender: TObject); rbBasicMD: TRadioButton; rbComposite: TRadioButton; rbFormLetter: TRadioButton; saveDialog1: Topenbalog, rbAboutclick(Sender: TObject); rbComposite: TRadioButton; rbFormLetter: TRadioButton; saveDialog1: Topenbalog, rbBasicMD: TRadioButton; rbFormLetter: TRadioButton; saveDialog1: Topenbalog, rbBasicMD: TRadioButton; rbFormLetter: TRadioButton; rbFormLetter: TRadioButton; saveDialog1: Topenbalog1: Topenbalog1: Topenbalog2: These saveDialog3: Topenbalog3: Topenb		•
rbFormLetterClick(Sender: TObject);  <> procedure btnExportClick(Sender: TObject);  <> procedure rbAboutClick(Sender: TObject);  <> procedure rbGroupingClick(Sender: TObject); private { Private declarations } FReport: TCustomQuickRep; CreateListReport: TQuickRep; <> procedure SetReport(Value: TCustomQuickRep);    Control of the view. TCombobox,   Label4: TLabel;   rbExprMemo: TRadioButton;   rbBasicMD: TRadioButton;   btnFilters: TButton;   rbNeedData: TRadioButton;   rbFormLetter: TRadioButton;   btnExport: TButton;   SaveDialog1: TSaveDialog;   rbAbout: TRadioButton;   rbGrouping: TRadioButton;   rbFormLetter: TRadioButton;   rbFormLe		
TObject);  <> procedure btnExportClick(Sender: TObject);  <> procedure rbAboutClick(Sender: TObject);  <> procedure rbGroupingClick(Sender: TObject); private { Private declarations } FReport: TCustomQuickRep; CreateListReport: TQuickRep; <> procedure SetReport(Value: TCustomQuickRep);    Cabet4. TEabet, rbExprMemo: TRadioButton; rbImage: TRadioButton; rbBasicMD: TRadioButton; btnFilters: TButton; rbComposite: TRadioButton; rbNeedData: TRadioButton; rbFormLetter: TRadioButton; saveDialog1: TSaveDialog; rbAbout: TRadioButton; rbGrouping: TRadi	-	
<pre> &lt;&gt; procedure btnExportClick(Sender: TObject); &lt;&gt; procedure rbAboutClick(Sender: TObject); &lt;&gt; procedure rbGroupingClick(Sender: TObject); private { Private declarations } FReport: TCustomQuickRep; CreateListReport: TQuickRep; &lt;&gt; procedure SetReport(Value: TCustomQuickRep); </pre> TDExpTMento: TRadioButton; rbImage: TRadioButton; btnFilters: TButton; rbComposite: TRadioButton; rbNeedData: TRadioButton; rbFormLetter: TRadioButton; SaveDialog1: TSaveDialog; rbAbout: TRadioButton; Image2: TImage;  > procedure	`	Label4: TLabel;
btnExportClick(Sender: TObject);  <> procedure rbAboutClick(Sender: TObject);  <> procedure rbGroupingClick(Sender: TObject);  private { Private declarations }  FReport: TCustomQuickRep;  CreateListReport: TQuickRep;  <> procedure SetReport(Value: TCustomQuickRep);    SetReportClick(Sender: TObject);   Tobject   Tobject     SetReport   Tobject     Tobject   TendioButton;   Tobject     Thimage: TRadioButton;   Thomage: TRadioButton;   Thomage: TRadioButton;   Thomage: TRadioButton;   Thomage: TRadioButton;   Tobject     Thomage: TRadioButton;   Thomage		rbExprMemo: TRadioButton;
btnExportClick(Sender: TObject);  <> procedure rbAboutClick(Sender: TObject);  <> procedure rbGroupingClick(Sender: TObject);  private { Private declarations }  FReport: TCustomQuickRep;  CreateListReport: TQuickRep;  <> procedure SetReport(Value: TCustomQuickRep);  rbBasicMD: TRadioButton; btnFilters: TButton; rbNeedData: TRadioButton; rbFormLetter: TRadioButton; SaveDialog1: TSaveDialog; rbAbout: TRadioButton; rbGrouping: TRadioButton; Image2: TImage;	1	rbImage: TRadioButton;
<pre>c&gt; procedure rbAboutClick(Sender: TObject);</pre>		
rbAboutClick(Sender: TObject); <pre></pre>	-	,
rbGroupingClick(Sender: TObject); private { Private declarations } FReport: TCustomQuickRep; CreateListReport: TQuickRep; <> procedure SetReport(Value: TCustomQuickRep);  Label5: TLabel; rbNeedData: TRadioButton; rbFormLetter: TRadioButton; SaveDialog1: TSaveDialog; rbAbout: TRadioButton; rbGrouping: TRadioButton; Image2: TImage;	` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` `	, , , , , , , , , , , , , , , , , , ,
rbGroupingClick(Sender: TObject); private { Private declarations } FReport: TCustomQuickRep; CreateListReport: TQuickRep; <> procedure SetReport(Value: TCustomQuickRep);  rbNeedData: TRadioButton; rbFormLetter: TRadioButton; SaveDialog1: TSaveDialog; rbAbout: TRadioButton; rbGrouping: TRadioButton; Image2: TImage;	-	•
ToustomQuickRep;   ToustomQuic	• ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` `	,
FREPORT: TCustomQuickRep;  CreateListReport: TQuickRep;  <> procedure SetReport(Value: TCustomQuickRep);  btnExport: TButton; SaveDialog1: TSaveDialog; rbAbout: TRadioButton; rbGrouping: TRadioButton; Image2: TImage;		,
CreateListReport: TQuickRep;  <> procedure SetReport(Value: TCustomQuickRep);  SaveDialog1: TSaveDialog; rbAbout: TRadioButton; rbGrouping: TRadioButton; Image2: TImage;	•	
CreateListReport: TQuickRep;  <> procedure SetReport(Value: TCustomQuickRep);  rbAbout: TRadioButton; rbGrouping: TRadioButton; Image2: TImage;		
SetReport(Value : rbGrouping: TRadioButton; TCustomQuickRep);  rbGrouping: TRadioButton; Image2: TImage;		
SetReport(Value : Image2: TImage;  TCustomQuickRep);	<> procedure	· ·
TCustomQuickRep);	SetReport(Value :	
public <> procedure	TCustomQuickRep);	
	public	<> procedure

from the table }	{ Public declarations }
SomeFields := TStringList.Create;	property Report :
with MyTable do	TCustomQuickRep read FReport
begin	write SetReport;
DatabaseName := 'DBDEMOS';	end;
TableName := 'COUNTRY.DB';	,
ReadOnly := True;	var
Active := True;	frmQR3Demo: TfrmQR3Demo;
1100210 ( 1100)	
// For this example, we will pull	implementation
the field names from the table	•
// If you wanted to only use some	uses mdrpt, exprmemo, image,
of the fields, you would edit	basicmd, needdata, frmltr,
// this list.	history, grouping;
for nIdx := 0 to FieldCount - 1 do	moory, grouping,
101 max. 0 to 1 leacount 1 do	{\$R *.dfm}
SomeFields.Add(Fields[nIdx].FieldN	(411 141111)
ame);	<> procedure
end;	TfrmQR3Demo.SetReport(Value :
ciu,	TCustomQuickRep);
// If you didn't create the report,	begin
you must set the report object to nil	FReport := Value;
// before calling QRCreateList	if Value $\Leftrightarrow$ nil then
// before canning QKC reateList	if Value =
avanant :- nile	
areport := nil;	TCustomQuickRep(QRCompositeR eport1) then
(Duild the veneut)	Description.Lines.Text :=
{ Build the report }	Composite_Description
// 16 1 41 - 1 141 - 14	else
// If you change the displaywidth, it	eise
will be reflecte in the created	Description.Lines.Assign(Report.Des
// report	
	cription);
with MyTable.Fields[1] do	end;
DisplayWidth := DisplayWidth	Z Sweepedowe
div 2;	<pre>&lt;&gt; procedure TfrmOD2Domo Crootel ist(Sondon)</pre>
	TfrmQR3Demo.CreateList(Sender:
// create the report	TObject);
QRCreateList(aReport, nil,	a Domonta T Custom Ouish Dom
MyTable, 'Country Listing',	aReport: TCustomQuickRep;
SomeFields);	SomeFields: TStringList;
	MyTable: TTable;
// Make the column header's font	nIdx: integer;
use bold attribute	begin
	{ Create a table on the fly, this
aReport.Bands.ColumnHeaderBand.	example uses a table from the demo
Font.Style := [fsBold];	database }
	<b>MyTable</b> := TTable.Create(self);
// Now adjust the spacing of the	
fields. There isn't any reason to	{ create the list of fields to output

end;	// do this, this is just to show how to
	access the controls on the
<> procedure	// report.
TfrmQR3Demo.QRCompositeRepor	1
t1AddReports(Sender: TObject);	for nIdx := 0 to
begin	aReport.Bands.ColumnHeaderBand.
// The OnAddReports event is	ControlCount -1 do
called by the CompositeReport	if
// to add the reports to list of	aReport.Bands.ColumnHeaderBand.
reports	Controls[nIdx] is TQRPrintable
with	then
QRCompositeReport1.Reports do	with
begin	TQRPrintable(aReport.Bands.Colu
begin	mnHeaderBand.Controls[nIdx]) do
Add(fumMastarDatail OvialrDan1).	Left := Left - (5 * nIdx);
Add(frmMasterDetail.QuickRep1);	Left Left - (5 max),
Add(frmBasicMD.QuickRep1);	e Ti o
Add(frmImageRpt.QuickRep1);	for $nIdx := 0$ to
end;	aReport.Bands.DetailBand.ControlC
end;	ount -1 do
,	if
<> procedure	aReport.Bands.DetailBand.Controls[
TfrmQR3Demo.btnCRClick(Sender	nIdx] is TQRPrintable then
	with
: TObject);	TQRPrintable(aReport.Bands.Detail
begin	
QRCompositeReport1.Preview;	Band.Controls[nIdx]) do
end;	Left := Left - (5 * nIdx);
<> procedure	
TfrmQR3Demo.rbCreateListClick(S	{ You can change the report objects
ender: TObject);	before calling the report }
begin	// areport.page.orientation :=
Report := CreateListReport;	poLandscape;
	{preview or print the report}
end;	(preview or print the report)
-	if sender = btnPreview then
<> procedure	
TfrmQR3Demo.rbMasterDetailClic	begin
k(Sender: TObject);	case cbPreview.ItemIndex of
begin	0: areport.preview;
Report :=	1: areport.previewModal;
frmMasterDetail.QuickRep1;	2: areport.previewModeless;
end;	end;
CHU,	end
	else if sender = btnPrint then
<> procedure	
TfrmQR3Demo.FormCreate(Sender	areport.print;
: TObject);	
var i: integer;	{ all done, free the objects }
begin	aReport.Free;
// Get the current QuickReport	MyTable.Free;
version number from	SomeFields.Free;
TOLSION NUMBER IN THE	Summer icius.i'i cc,

ODCompositeDeposet1 prints	// the commute unit
QRCompositeReport1.print;	// the qrprntr unit
end	i := pos(' ', cQRName);
else	if i >=0 then
begin	VersionLbl.Caption := 'Version' +
if sender = btnPreview then	copy(cQRName, i, 99);
begin	// Create the report object used by
case cbPreview.ItemIndex of	the QRCreateList() function
0: report.preview;	CreateListReport :=
1: report.previewModal;	TQuickRep.Create(self);
2: report.previewModeless;	CreateListReport.Description.Text
end;	:= 'Example of how to call the
end	QRCreateList function';
else if sender = btnPrint then	cbPreview.ItemIndex := 0;
report.print;	end;
end;	
end;	<> procedure
end;	TfrmQR3Demo.FormActivate(Send
	er: TObject);
<> procedure	begin
TfrmQR3Demo.rbExprMemoClick(	if Description.Lines.Count = 0 then
Sender: TObject);	begin
begin	rbCreateListClick(Self);
Report :=	rbCreateList.Checked := True;
frmExprmemo.QuickRep1;	end;
end;	end;
	,
<> procedure	<> procedure
TfrmQR3Demo.rbImageClick(Send	TfrmQR3Demo.btnPreviewClick(Se
er: TObject);	nder: TObject);
begin	begin
Report :=	// This code is more complicated
frmImageRpt.QuickRep1;	than what you would
end;	// typically need to run a report.
chu,	Most of this code
<> procedure	// is to handle the selection of the
TfrmQR3Demo.rbBasicMDClick(Se	various types of
nder: TObject);	// reports.
begin	100100
	if report <> nil then
Report := frmRasicMD QuickPan1:	begin
frmBasicMD.QuickRep1;	if report = CreateListReport then
end;	CreateList(Sender)
Z	else if report =
<pre>&lt;&gt; procedure Tfrm OB2Dama htm Eilteau Click(San)</pre>	TCustomQuickRep(QRCompositeR
TfrmQR3Demo.btnFiltersClick(Sen	
der: TObject);	eport1) then
begin No. (NY)	begin
MessageDlg('When an export filter	if sender = btnPreview then
control is dropped on a form, it''s	QRCompositeReport1.preview
added to all of the	else

ASCII: TQRAsciiExportFilter
CSV: TQRCommaSeparatedFilter
In Professional Version:
RTF: TQRRTFExportFilter
WMF: TQRWMFExportFilter
Excel: TQRXLSFilter
}
end;
end;
btnExport.Enabled := True;
end;
,
<> procedure
TfrmQR3Demo.rbAboutClick(Send
er: TObject);
begin
Report := frmHistory.QuickRep1;
end;
<> procedure
TfrmQR3Demo.rbGroupingClick(Se
nder: TObject);
begin
Report :=
frmGrouping.QuickRep1;
end;
end.
111111111111111111111111111111111111111

<> Objects83
الهلف النصي لبرناهج
(Qr3)
ا عداد
علاء الدين اللباد
واتف ۷۱۳۵۷۵۶۹۰
<> Object frmQR3Demo:
TfrmQR3Demo
Left = 182
Top = 108
Width = 603
Height = 340
Caption = 'Quick Report 3

<pre>previews',mtInformation, [mbok], 0);</pre>
end;
enu,
<> procedure
TfrmQR3Demo.rbCompositeClick(S
ender: TObject);
begin
Report :=
TCustomQuickRep(QRCompositeR
eport1);
end;
<> procedure
TfrmQR3Demo.rbNeedDataClick(S
ender: TObject);
begin
Report :=
frmNeedData.QuickRep1;
end;
Z Same a Jame
<pre>&lt;&gt; procedure TformOD2Dome ub Form LetterClick()</pre>
TfrmQR3Demo.rbFormLetterClick( Sender: TObject);
begin
Report :=
frmFormLetter.QuickRep1;
end;
· · ·
// The following code show how to
explicitly call an export
// filter without going through the
preview
<> procedure
TfrmQR3Demo.btnExportClick(Sen
der: TObject);
begin
btnExport.Enabled := False;
with SaveDialog1 do
begin
if Execute then
begin
frmFormLetter.QuickRep1.ExportT
oFilter(TQRCommaSeparatedFilter.
Create(FileName));
(College Cilescon
Other filters:
HTML: TQRHTMLDocumentFilter

XX': 141 270
Width = 279
Height = 13
Caption = 'Copyright 1995 - 1999
QuSoft AS http://www.qusoft.com'
Transparent = True
end
<> Object Label4: TLabel
Left = 24
Top = 272
Width = 61
Height = 13
Alignment = taRightJustify
Caption = 'Preview type'
end
<> Object Label5: TLabel
Left = 56
Top = 50
Width = 158
Height = 13
Caption = 'Demo version 3.1.0
July 13, 1999'
Transparent = True
<> Object Image2: TImage Left = 6
Top = 6
Width = 47
Height = 47
Picture.Data = {
Transparent = True
end
<> Object GroupBox1:
TGroupBox
Left = 7
Top = 64
Width = 586
Height = 193
Caption = 'Select a Report'
TabOrder = 0
<> Object rbCreateList:
TRadioButton
Left = 4
Top = 32 Width = 179
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Height = 17
Caption = 'QRCreatelist()
example'

Standard Demo for Delphi 6'
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
Scaled = False
OnActivate = FormActivate
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> Object Label1: TLabel
Left = 56
Top = 4
Width = 174
Height = 34
Caption = 'QuickReport'
Font.Charset =
DEFAULT_CHARSET  Font.Color = clBlack
Font.Height = -29
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
Transparent = True
end
<> Object VersionLbl: TLabel
Left = 236
Top = 18
Width = 70
Height = 16
Caption = 'Version 3.0'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
Transparent = True
end
<> Object Label3: TLabel
Left = 56
Top = 36
10p 00

Top = 49
Width = 197
Height = 17
Caption = 'Basic Master/Detail
example'
TabOrder = 3
OnClick = rbBasicMDClick
end
<> Object rbComposite:
TRadioButton
Left = 4
Top = 135
Width = 179
Height = 17
Caption = 'Composite Report'
TabOrder = 8
OnClick = rbCompositeClick
end
<> Object rbNeedData:
TRadioButton
Left = 4
Top = 152
Width = 179
Height = 17
Caption = 'OnNeedData
=
example' TabOrder = 9
OnClick = rbNeedDataClick
end
<> Object rbFormLetter:
TRadioButton
Left = 4
Top = 118
Width = 179
Height = 17
Caption = 'Form Letter example'
TabOrder = 7
OnClick = rbFormLetterClick
end
<> Object rbAbout:
TRadioButton
Left = 4
Top = 15
Width = 179
Height = 17
Caption = 'QR3DEMO History'
TabOrder = 1
OnClick = rbAboutClick

TabOrder = 2
OnClick = rbCreateListClick
end
<> Object rbMasterDetail:
TRadioButton
Left = 4
Top = 66
Width = 179
Height = 17
Caption = 'Master/Detail
example'
TabOrder = 4
OnClick = rbMasterDetailClick
end
<> Object Description:
TMemo
Left = 207
Top = 16
Width = 370
Height = 153
ReadOnly = True
ScrollBars = ssVertical
TabOrder = 0
end
<> Object rbExprMemo:
TRadioButton
Left = 4
Top = 101 Width = 179
1 1 2 2 2
Height = 17
Caption = 'ExprMemo example'
TabOrder = 6
OnClick = rbExprMemoClick
end
<> Object rbImage:
TRadioButton
Left = 4
Top = 83
Width = 179
Height = 17
Caption = 'List report with
bitmaps'
TabOrder = 5
OnClick = rbImageClick
end
<> Object rbBasicMD:
TRadioButton
Left = 4

Left = 392
Top = 5
Width = 113
Height = 25
Caption = 'About the filters'
TabOrder = 4
OnClick = btnFiltersClick
end
<> Object btnExport: TButton
Left = 488
Top = 268
Width = 75
Height = 25
Caption = 'CSV Export'
TabOrder = 5
OnClick = btnExportClick
end
<> Object OpenDialog1:
TOpenDialog
DefaultExt = 'TXT'
Filter = 'Text files (*.txt) *.TXT'
Left = 240
Top = 155
end
<> Object SaveDialog1:
TSaveDialog
DefaultExt = 'RTF'
Filter = 'Richtext files (*.rtf) *.rtf'
Title = 'Export RTF'
Left = 240
Top = 91
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
2
<> Object frmNeedData:
TfrmNeedData
Left = 192
Top = 107
Width = 696
Height = 480
Caption = 'frmNeedData'
Color = clBtnFace
Font.Charset =
Font.Cnarset = DEFAULT CHARSET
<del>_</del>
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'

end
<> Object rbGrouping:
TRadioButton
Left = 4
Top = 169
Width = 179
Height = 17
Caption = 'Grouping and
subtotaling'
TabOrder = 10
OnClick = rbGroupingClick
end
end
<> Object btnPreview:
TButton
Left = 360
Top = 268
Width = 59
Height = 25
Caption = 'Preview'
TabOrder = 1
OnClick = btnPreviewClick
end
<> Object btnPrint: TButton
Left = 424
Top = 268
Width = 57
Height = 25
Caption = 'Print'
•
TabOrder = 2
OnClick = btnPreviewClick
end
<> Object cbPreview:
TComboBox
Left = 89
Top = 268
Width = 265
Height = 21
Style = csDropDownList
ItemHeight = 13
TabOrder = 3
Items.Strings = (
'Standard modeless preview
(non-threaded)'
'Modal (runs in a thread)'
'Modeless (runs in a thread)')
end
<> Object btnFilters: TButton

SQL.Strings = (
'select * from orders'
'where CustNo = :CustNo'
'order by CustNo, OrderNo')
Left = 48
ParamData = <
item
DataType = ftFloat
Name = 'CustNo'
ParamType = ptUnknown
Size = 8
end>
end
<> Object dsOrders:
TDataSource
DataSet = qryOrders
Top = 27
end
<> Object qryItems: TQuery Active = True
DatabaseName = 'DBDEMOS'
DataSource = dsOrders
SQL.Strings = (
'SELECT * '
'FROM items'
'WHERE OrderNo' = :OrderNo'
'ORDER BY OrderNo, ItemNo')
Left = 72
ParamData = <
item
DataType = ftFloat
Name = 'OrderNo'
ParamType = ptUnknown
Size = 8
end>
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
4
<> Object frmImageRpt:
TfrmImageRpt
Left = 192
Top = 107
Width = 696
Height = 480
Caption = 'frmImageRpt'
. 8. 1.

T
Font.Style = []
OldCreateOrder = False
Scaled = False
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
3
<> Object frmMasterDetail:
TfrmMasterDetail
Left = 153
Top = 119
Width = 799
Height = 480
Caption = 'frmMasterDetail'
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -11 Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Scaled = False
PixelsPerInch = 96
TextHeight = 13
<> Object dsCustomer:
TDataSource
DataSet = qryCustomer
end
<> Object qryCustomer:
TQuery
Active = True
DatabaseName = 'DBDEMOS'
SQL.Strings = (
'select * from customer'
'Where CustNo <= 1354'
'order by CustNo')
Left = 24
end
<> Object qryOrders: TQuery
Active = True
DatabaseName = 'DBDEMOS'
DataSource = dsCustomer
DataSource - usCustomer

Size = 50
end
<> Object tbBioGraphic:
TGraphicField
FieldName = 'Graphic'
BlobType = ftGraphic
end
end
End
\\\\\\\
5
<> Object frmHistory:
TfrmHistory
Left = 192
Top = 107
Width = 696
Height = 480
Caption = 'frmHistory'
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = False
Scaled = False
PixelsPerInch = 96
TextHeight = 13
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
6
<> Object frmGrouping:
TfrmGrouping
Left = 192
Top = 107
Width = 774
Height = 480
Caption = 'frmGrouping'
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'

Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = False
Scaled = False
PixelsPerInch = 96
TextHeight = 13
<> Object tbBio: TTable
Active = True
DatabaseName = 'DBDEMOS'
TableName = 'BIOLIFE.DB'
Left = 8
Top = 16
<> Object tbBioSpeciesNo:
TFloatField
FieldName = 'Species No'
end
<> Object tbBioCategory:
TStringField
FieldName = 'Category'
Size = 15
end
<> Object
tbBioCommon Name: TStringField
FieldName = 'Common Name'
Size = 30
end
<> Object tbBioSpeciesName:
TStringField  FieldName = !Sneeies Name!
FieldName = 'Species Name'
Size = 40
end
<> Object tbBioLengthcm:
TFloatField
FieldName = 'Length (cm)'
end
<pre>&lt;&gt; Object tbBioLength_In:</pre>
TFloatField
FieldName = 'Length_In'
end
<> Object tbBioNotes:
TMemoField
FieldName = 'Notes'
BlobType = ftMemo

<> Object DetailQry: TQuery
Active = True
DatabaseName = 'DBDEMOS'
DataSource = DataSource1
SQL.Strings = (
'SELECT OrderNo, Qty,
ItemNo'
'FROM ITEMS I'
'WHERE (i.OrderNo =
:OrderNo)'
'ORDER BY OrderNo, ItemNo')
Left = 8
Top = 64
ParamData = <
item
DataType = ftFloat
Name = 'orderno'
ParamType = ptUnknown
Size = 8
end>
<> Object
DetailQryOrderNo: TFloatField
FieldName = 'OrderNo'
Origin =
'"ITEMS.DB".OrderNo'
end
<> Object DetailQryQty:
TIntegerField
FieldName = 'Qty'
Origin = ""ITEMS.DB".Qty"
end
<> Object DetailQryItemNo:
TFloatField
FieldName = 'ItemNo'
Origin = ""ITEMS.DB".ItemNo"
end
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
7
<> Object frmFormLetter:
TfrmFormLetter
Left = 200
Top = 108
Width = 544
Height = 375

Font.Style = []
OldCreateOrder = False
Scaled = False
PixelsPerInch = 96
TextHeight = 13
U
<> Object MasterQry: TQuery
Active = True
DatabaseName = 'DBDEMOS'
SQL.Strings = (
'select c.company, o.orderno,
o.custno, o.ItemsTotal'
'from orders o join customer c on
o.custno=c.custno'
'where c.company LIKE "A%""
'order by c.company, o.orderno')
Left = 8
<> Object
MasterQrycompany: TStringField
FieldName = 'company'
Origin =
"CUSTOMER.DB".Company
Size = 30
end
<> Object
MasterQryorderno: TFloatField
FieldName = 'orderno'
Origin =
"ORDERS.DB".OrderNo'
end
<> Object MasterQrycustno:
TFloatField
FieldName = 'custno'
Origin =
"ORDERS.DB".CustNo'
end
<> Object
MasterQryItemsTotal:
TCurrencyField
FieldName = 'ItemsTotal'
Origin =
S
"ORDERS.DB".ItemsTotal
end
end
<> Object DataSource1:
<b>TDataSource</b>
DataSet = MasterQry
Left = 8
Top = 32
end
CHU

end
<> Object
qryEmployeePhoneExt:
TStringField
FieldName = 'PhoneExt'
Origin =
'"Employee.DB".PhoneExt'
Size = 4
end
<> Object
qryEmployeeHireDate:
<b>TDateTimeField</b>
FieldName = 'HireDate'
Origin =
'"Employee.DB".HireDate'
end
<> Object
qryEmployeeSalary: TFloatField
CustomConstraint = 'X > 4499'
ConstraintErrorMessage =
'Minimum Salary is \$4,500.00'
FieldName = 'Salary'
Origin =
'"Employee.DB".Salary'
currency = True
end
end
End
111111111111111111111111111111111111111
8
<> Object frmExprmemo:
TfrmExprmemo
Left = 192
Top = 107
Width = 696
Height = 480
Caption = 'frmExprmemo'
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = False
Scaled = False
PixelsPerInch = 96
TextHeight = 13

Caption = 'frmFormLetter'
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Scaled = False
PixelsPerInch = 96
TextHeight = 13
<> Object qryEmployee:
TQuery
Active = True
DatabaseName = 'DBDEMOS'
SQL.Strings = (
'SELECT *'
'FROM Employee'
'WHERE HireDate <
"01/01/1990"")
Left = 8
Top = 8
<> Object
qryEmployeeEmpNo: TIntegerField
CustomConstraint = 'Value > 0'
ConstraintErrorMessage =
'EmpNo cannot be 0 or negative'
_
'EmpNo cannot be 0 or negative'
'EmpNo cannot be 0 or negative' FieldName = 'EmpNo' Origin =
'EmpNo cannot be 0 or negative' FieldName = 'EmpNo'
'EmpNo cannot be 0 or negative' FieldName = 'EmpNo' Origin = '"Employee.DB".EmpNo' DisplayFormat = 'Emp''#'' 0000'
'EmpNo cannot be 0 or negative' FieldName = 'EmpNo' Origin = '"Employee.DB".EmpNo'
'EmpNo cannot be 0 or negative' FieldName = 'EmpNo' Origin = '"Employee.DB".EmpNo' DisplayFormat = 'Emp''#'' 0000' MaxValue = 9999 MinValue = 1
'EmpNo cannot be 0 or negative' FieldName = 'EmpNo' Origin = '"Employee.DB".EmpNo' DisplayFormat = 'Emp''#'' 0000' MaxValue = 9999 MinValue = 1 end
'EmpNo cannot be 0 or negative' FieldName = 'EmpNo' Origin = '"Employee.DB".EmpNo' DisplayFormat = 'Emp''#'' 0000' MaxValue = 9999 MinValue = 1 end <> Object
'EmpNo cannot be 0 or negative' FieldName = 'EmpNo' Origin = '"Employee.DB".EmpNo' DisplayFormat = 'Emp''#'' 0000' MaxValue = 9999 MinValue = 1 end <> Object qryEmployeeLastName:
'EmpNo cannot be 0 or negative' FieldName = 'EmpNo' Origin = '"Employee.DB".EmpNo' DisplayFormat = 'Emp''#'' 0000' MaxValue = 9999 MinValue = 1 end <> Object qryEmployeeLastName: TStringField
'EmpNo cannot be 0 or negative'  FieldName = 'EmpNo'  Origin =  '"Employee.DB".EmpNo'  DisplayFormat = 'Emp''#'' 0000'  MaxValue = 9999  MinValue = 1  end  <> Object qryEmployeeLastName: TStringField  FieldName = 'LastName'
'EmpNo cannot be 0 or negative' FieldName = 'EmpNo' Origin = '"Employee.DB".EmpNo' DisplayFormat = 'Emp''#'' 0000' MaxValue = 9999 MinValue = 1 end <> Object qryEmployeeLastName: TStringField FieldName = 'LastName' Origin =
'EmpNo cannot be 0 or negative' FieldName = 'EmpNo' Origin = '"Employee.DB".EmpNo' DisplayFormat = 'Emp''#'' 0000' MaxValue = 9999 MinValue = 1 end <> Object qryEmployeeLastName: TStringField FieldName = 'LastName' Origin = '"Employee.DB".LastName'
'EmpNo cannot be 0 or negative' FieldName = 'EmpNo' Origin = '"Employee.DB".EmpNo' DisplayFormat = 'Emp''#'' 0000' MaxValue = 9999 MinValue = 1 end <> Object qryEmployeeLastName: TStringField FieldName = 'LastName' Origin = '"Employee.DB".LastName' end
'EmpNo cannot be 0 or negative' FieldName = 'EmpNo' Origin = '"Employee.DB".EmpNo' DisplayFormat = 'Emp''#'' 0000' MaxValue = 9999 MinValue = 1 end <> Object qryEmployeeLastName: TStringField FieldName = 'LastName' Origin = '"Employee.DB".LastName' end <> Object
'EmpNo cannot be 0 or negative' FieldName = 'EmpNo' Origin = '"Employee.DB".EmpNo' DisplayFormat = 'Emp''#'' 0000' MaxValue = 9999 MinValue = 1 end <> Object qryEmployeeLastName: TStringField FieldName = 'LastName' Origin = '"Employee.DB".LastName' end <> Object qryEmployeeFirstName:
'EmpNo cannot be 0 or negative' FieldName = 'EmpNo' Origin = '"Employee.DB".EmpNo' DisplayFormat = 'Emp''#'' 0000' MaxValue = 9999 MinValue = 1 end <> Object qryEmployeeLastName: TStringField FieldName = 'LastName' Origin = '"Employee.DB".LastName' end <> Object qryEmployeeFirstName: TStringField
'EmpNo cannot be 0 or negative' FieldName = 'EmpNo' Origin = '"Employee.DB".EmpNo' DisplayFormat = 'Emp''#'' 0000' MaxValue = 9999 MinValue = 1 end <> Object qryEmployeeLastName: TStringField FieldName = 'LastName' Origin = '"Employee.DB".LastName' end <> Object qryEmployeeFirstName: TStringField FieldName = 'FirstName'
'EmpNo cannot be 0 or negative' FieldName = 'EmpNo' Origin = '"Employee.DB".EmpNo' DisplayFormat = 'Emp''#'' 0000' MaxValue = 9999 MinValue = 1 end <> Object qryEmployeeLastName: TStringField FieldName = 'LastName' Origin = '"Employee.DB".LastName' end <> Object qryEmployeeFirstName: TStringField FieldName = 'FirstName' Origin =
'EmpNo cannot be 0 or negative' FieldName = 'EmpNo' Origin = '"Employee.DB".EmpNo' DisplayFormat = 'Emp''#'' 0000' MaxValue = 9999 MinValue = 1 end <> Object qryEmployeeLastName: TStringField FieldName = 'LastName' Origin = '"Employee.DB".LastName' end <> Object qryEmployeeFirstName: TStringField FieldName = 'FirstName'

FieldName = 'Country' end <> Object tbCustomerPhone: TStringField FieldName = 'Phone'
<> Object tbCustomerPhone: TStringField
TStringField
TStringField
FieldName = 'Phone'
Size = 15
end
<> Object tbCustomerFAX:
TStringField
FieldName = 'FAX'
Size = 15
end
<> Object
tbCustomerTaxRate: TFloatField
FieldName = 'TaxRate'
DisplayFormat = '0.00%'
end
<> Object
tbCustomerContact: TStringField
FieldName = 'Contact'
FieldName = 'Contact' end
end
end <> Object
end
end <> Object tbCustomerLastInvoiceDate:
end <> Object tbCustomerLastInvoiceDate: TDateTimeField
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate'
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end End
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end End  [
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end End
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end End
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end End  Lind  9 <> Object frmBasicMD: TfrmBasicMD
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end End
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end End
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end End  William = 'LastInvoiceDate' end End  Left = 192 Top = 107 Width = 696
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end End  William = 'LastInvoiceDate' end End  Left = 192 Top = 107 Width = 696 Height = 480
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end End  William = 'LastInvoiceDate' end End  FieldName = 'LastInvoiceDate' end End  End  William = 'LastInvoiceDate' end End  End  William = 'Gasting = 'Gast
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end End
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end End  End  9 <> Object frmBasicMD: TfrmBasicMD Left = 192 Top = 107 Width = 696 Height = 480 Caption = 'frmBasicMD' Color = clBtnFace Font.Charset =
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end End  William = 'LastInvoiceDate' end End  Second = 'LastInvoiceDate' end End  William = 'LastInvoiceDate' end End  William = 'LastInvoiceDate' end End  End  Second = 'Instance
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end End  Second Seco
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end End  Section = 'Independent of the state of the s
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end End  Second Seco
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end End  Second End  William End  Second En
end <> Object tbCustomerLastInvoiceDate: TDateTimeField FieldName = 'LastInvoiceDate' end end End  Second Seco

<> Object tbCustomer:
TTable
Active = True
DatabaseName = 'DBDEMOS'
IndexFieldNames = 'CustNo'
TableName = 'CUSTOMER.DB'
Left = 32
Top = 24
<> Object
tbCustomerCustNo: TFloatField
Alignment = taLeftJustify
FieldName = 'CustNo'
DisplayFormat = 'CN 0000'
MaxValue =
9999.0000000000000000000000000000000000
MinValue =
1000.000000000000000000
end
<> Object
tbCustomerCompany: TStringField
FieldName = 'Company'
Size = 30
end
<> Object tbCustomerAddr1:
TStringField
FieldName = 'Addr1'
Size = 30
end
<> Object tbCustomerAddr2:
TStringField
FieldName = 'Addr2'
Size = 30
end
<> Object tbCustomerCity:
TStringField
FieldName = 'City'
Size = 15
end
<> Object tbCustomerState:
TStringField
FieldName = 'State'
end
<> Object tbCustomerZip:
TStringField
FieldName = 'Zip'
Size = 10
end
tbCustomerCountry: TStringField
<> Object tbCustomerCountry: TStringField

PixelsPerInch = 96
TextHeight = 13
<> Object dsCustomer:
<b>TDataSource</b>
DataSet = tbCustomer
Left = 24
end
<> Object tbCustomer:
TTable
Active = True
DatabaseName = 'DBDEMOS'
IndexFieldNames = 'Company'
TableName = 'CUSTOMER.DB'
Left = 24
Top = 32
end
<> Object qryOrders: TQuery
Active = True
DatabaseName = 'DBDEMOS'
DataSource = dsCustomer
SQL.Strings = (
'select * from orders '
'WHERE (CustNo = :CustNo)'
'order by custno, Terms')
Left = 24
Top = 56
ParamData = <
item
DataType = ftFloat
Name = 'CustNo'
ParamType = ptUnknown
Size = 8
end>
end
End
\\\\\\\\\
10

{ Public declarations }
end;
var
Form1: TForm1;
,
implementation
•
{\$R *.DFM}
<> procedure
TForm1.BSMorphButton1Click(Sen
der: TObject);
begin
ShowMessage (' '
+(Sender as
TBSMorphButton).Hint
+' ');
end;
<> procedure
TForm1.BSMorphButton5Click(Sen
der: TObject);
begin
Close;
end;
end.

<> Object\$5
الهلف النصي لبرناهج
(BS Morph Button)
عامدا
علاء الدين اللباد
هاتف ۷۳۷۱۹۰۱
<> Object Form1: TForm1
Left = 158
Top = 159
BorderStyle = bsToolWindow
Caption = 'BSMorphButton
DEMO'
ClientHeight = 129
ClientWidth = 520

القسم الثالث
برامج منوعة في دلفي
Procedures 4
الهلف التنفيذي لبرنامج
(BS Morph Button)
عامدا
علاء الدين اللباد
هاتف ۷۹۵۷۵۲۹۰
unit Unit1;
,
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
BSMorphButton, StdCtrls;
type
TForm1 = class(TForm)
BSMorphButton1:
TBSMorphButton;
Label1: TLabel; BSMorphButton2:
TBSMorphButton;
Label2: TLabel;
BSMorphButton3:
TBSMorphButton;
Label3: TLabel;
Label4: TLabel;
BSMorphButton4:
TBSMorphButton;
BSMorphButton5:
TBSMorphButton;
<pre>&lt;&gt; procedure BSMorphButton1Click(Sender:</pre>
TObject);
<> procedure
BSMorphButton5Click(Sender:
TObject);
private
{ Private declarations }
public

Width = 120
Height = 13
Alignment = taCenter
AutoSize = False
Caption = 'L I G H T S'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWhite
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
Transparent = True
end
<> Object Label4: TLabel
Left = 392
Top = 9
Width = 120
Height = 13
Alignment = taCenter
AutoSize = False
Caption = 'M O V E'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWhite
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
Transparent = True
end
End

C I IDI I
Color = clBlack
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = False
Position = poScreenCenter
PixelsPerInch = 96
TextHeight = 13
<> Object Label1: TLabel
Left = 8
Top = 9
Width = 120
Height = 13
Alignment = taCenter
AutoSize = False
Caption = 'S H A D O W'
Font.Charset =
DEFAULT CHARSET
Font.Color = clWhite
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
Transparent = True
end
<> Object Label2: TLabel
Left = 136
Top = 9
Width = 120
Height = 13
Alignment = taCenter
AutoSize = False
Caption = 'C O L O R S'
Font.Charset =
DEFAULT CHARSET
Font.Color = clWhite
Font. Height = -11
Font Style = II
Font.Style = [] ParentFont = False
Transparent = True
end
<> Object Label3: TLabel
Left = 264
Top = 9

Label5: TLabel;
Label4: TLabel;
Label3: TLabel;
cmbxMode: TComboBox;
Button1: TButton;
Bevel3: TBevel;
chkbxAbortKey: TCheckBox;
chkbxAbortProblem: TCheckBox;
chkbxTrans: TCheckBox;
Label1: TLabel;
·
Label2: TLabel;
Label10: TLabel;
Label9: TLabel;
Label12: TLabel;
edtChangedTable: TEdit;
Label13: TLabel;
edtKeyVioTbl: TEdit;
Label14: TLabel;
edtProbTbl: TEdit;
Label11: TLabel;
edtRecCount: TEdit;
<>procedure
FormCreate(Sender: TObject);
<>procedure
cmbxSourceAliasChange(Sender:
<u> </u>
TObject); <>procedure
_
cmbxDestAliasChange(Sender:
TObject);
<>procedure
cmbxSourceTableChange(Sender:
TObject);
<>procedure
cmbxDestTableChange(Sender:
TObject);
<>procedure
cmbxSourceIndexChange(Sender:
TObject);
<>procedure
cmbxDestIndexChange(Sender:
TObject);
<>procedure
Button1Click(Sender: TObject);
<>procedure
chkbxAbortKeyClick(Sender:
TObject);
<>procedure
chkbxAbortProblemClick(Sender:
TObject);
1 Objects,

الملف التنفيذي لبرنامج	
(BatchMv)	
عامدا	
علاء الدين اللباد	
واتف ۷۳۵۷۵۶۹۰	
	_
Procedures 28	
unit Batmove;	
interface	
uses	
Windows, Messages, SysUtils,	
Classes, Graphics, Controls, Forms,	
Dialogs,	
StdCtrls, ExtCtrls, Db, DBTables,	
ComCtrls;	
type	_
TForm1 = class(TForm)	
BatchMove1: TBatchMove;	_
tblSource: TTable;	
tblDest: TTable;	
Bevel1: TBevel;	
Label6: TLabel;	
cmbxDestAlias: TComboBox;	
Label7: TLabel;	
cmbxDestTable: TComboBox;	
Label8: TLabel;	
cmbxDestIndex: TComboBox;	
Bevel2: TBevel;	
cmbxSourceAlias: TComboBox;	
cmbxSourceTable: TComboBox;	
cmbxSourceIndex: TComboBox;	

	<>procedure
Session.GetTableNames(tblSource.D	chkbxTransClick(Sender: TObject);
atabaseName,",true,false,cmbxSour	<>procedure
ceTable.Items);	cmbxModeChange(Sender:
end	TObject);
else	<>procedure
begin	edtRecCountKeyPress(Sender:
tblSource.DatabaseName := ";	TObject; var Key: Char);
cmbxSourceTable.Items.Clear;	<>procedure
end;	FormShow(Sender: TObject);
end;	private
	{ Private declarations }
<>procedure	function IsStringsEqual(const
TForm1.cmbxDestAliasChange(Sen	s1,s2 : string): boolean;
der: TObject);	// simple utility function
begin	public
if cmbxDestAlias.ItemIndex <> -1	{ Public declarations }
then	end;
begin	
tblDest.DatabaseName :=	var
cmbxDestAlias.Items[cmbxDestAlias	Form1: TForm1;
.ItemIndex];	
	implementation
Session.GetTableNames(tblDest.Dat	
abaseName,",true,false,cmbxDestTa	<b>{\$R *.DFM}</b>
ble.Items);	
end	<>procedure
else	TForm1.FormCreate(Sender:
begin	TObject); // Get alias names
tblDest.DatabaseName := ";	begin
cmbxDestTable.Items.Clear;	if Session.Active = false then
end;	Session.Open;
end;	
	Session.GetAliasNames(CmbxSourc
<>procedure	eAlias.Items);
TForm1.cmbxSourceTableChange(S	cmbxDestAlias.Items :=
ender: TObject);	cmbxSourceAlias.Items;
begin	end;
if cmbxSourceTable.ItemIndex <> -	
1 then	<>procedure
begin	TForm1.cmbxSourceAliasChange(S
tblSource.TableName :=	ender: TObject);
cmbxSourceTable.Items[cmbxSourc	begin
eTable.ItemIndex];	if cmbxSourceAlias.ItemIndex <> -
thlSource CotIndevNemes(ambySou	1 then
tblSource.GetIndexNames(cmbxSou rceIndex.Items);	begin
end	tblSource.DatabaseName :=
	cmbxSourceAlias.Items[cmbxSource
else	Alias.ItemIndex];

if cmbxDestIndex.ItemIndex <> -1	begi
then	tbl
begin	cm
tblDest.IndexName :=	end;
cmbxDestIndex.Items[cmbxDestInde	
x.ItemIndex];	end;
end	
else	<
begin	TFor
tblDest.IndexName := ";	der:
end;	begin
	if cn
end;	then
	begi
<>procedure	tb
TForm1.Button1Click(Sender:	cmbx
TObject);	le.Ite
begin	10.116
if tblDest.TableName = " then	tblDe
tblDest.TableName :=	dex.I
	end
cmbxDestTable.Text;	else
if ((tblSource.DatabaseName <> '') and // test for enough input	-
<u> </u>	begi tbl
(tblSource.TableName <> '') and	
(tblDest.DatabaseName <> '') and	cm
(tblDest.TableName <> '') and	end
	end;
(cmbxMode.items[cmbxMode.ItemI	
ndex] <> '')) then	<
begin	TFor
BatchMovel.ChangedTableName	ender
:= edtChangedTable.Text; // more	begin
batchmove setup	if cn
BatchMove1.KeyViolTableName	1 the
:= edtKeyVioTbl.Text;	begi
BatchMove1.ProblemTableName	tbl
:= edtProbTbl.Text;	cmbx
BatchMove1.RecordCount :=	eInde
StrToInt(edtRecCount.Text);	end
end	else
else	begi
begin	tb
Degin	
MessageDlg('Incomplete	end
G	end;
MessageDlg('Incomplete	
MessageDlg('Incomplete input.',mtError,[mbOK],0);	end;
MessageDlg('Incomplete input.',mtError,[mbOK],0); exit;	end;
MessageDlg('Incomplete input.',mtError,[mbOK],0); exit; end;	end; end; < TFor der:

begin
tblSource.TableName := '';
cmbxSourceIndex.Items.Clear;
end;
end;
< >mmooodumo
<pre>&lt;&gt;procedure TForm1.cmbxDestTableChange(Sen</pre>
der: TObject);
begin
if cmbxDestTable.ItemIndex <> -1
then
begin
tblDest.TableName :=
cmbxDestTable.Items[cmbxDestTab
le.ItemIndex];
ic.iciiiiiucaj,
tblDest.GetIndexNames(cmbxDestIn
dex.Items);
end
else
begin
tblDest.TableName := ";
cmbxDestIndex.Items.Clear;
end;
end;
<>procedure
TForm1.cmbxSourceIndexChange(S
ender: TObject);
begin
if cmbxSourceIndex.ItemIndex <> -
1 then
begin
tblSource.IndexName :=
cmbxSourceIndex.Items[cmbxSourc
eIndex.ItemIndex];
end
else
begin
tblSource.IndexName := ";
end;
end;
<>procedure
TForm1.cmbxDestIndexChange(Sen
der: TObject);
begin
~~8***

sStringsEqual(cmbxMode.Items[cm oxMode.ItemIndex],'Copy') then '+In	mber of records applied: ntToStr(BatchMove1.MovedCou mtInformation,[mbOK],0);
oxMode.ItemIndex],'Copy') then nt),	mtInformation,[mbOK],0);
Datah Mayat Mada hat Cany	:
BatchMove1.Mode := batCopy end	• 7
else if	
sStringsEqual(cmbxMode.Items[cm <	>procedure
· · · · · · · · · · · · · · · · · · ·	orm1.chkbxAbortKeyClick(Send
	TObject);
BatchMove1.Mode := begin	
	atchMove1.AbortOnKeyViol :=
	bxAbortKey.Checked;
sStringsEqual(cmbxMode.Items[cm end	· ·
oxMode.ItemIndex],'Delete') then	,
	>procedure
	orm1.chkbxAbortProblemClick(S
	The state of the s
oxMode.ItemIndex],'Update') then begin	er: TObject);
	ntchMove1.AbortOnProblem :=
	bxAbortProblem.Checked;
MessageDlg('Batch mode not end	<u>;</u>
ound',mtError,[mbOK],0);	
	>procedure
	orm1.chkbxTransClick(Sender:
	bject);
beg	
V VI	tchMove1.Transliterate :=
	bxTrans.Checked;
Form1.edtRecCountKeyPress(Sen end	;
ler: TObject; var Key: Char);	
	ction
	orm1.IsStringsEqual(const s1,s2:
	ng): boolean;
key := #0; begi	in
nd; Re	esult := UpperCase(s1) =
Upp	perCase(s2);
<pre>&lt;&gt;procedure end</pre>	;
Form1.FormShow(Sender:	
(Object);	
oegin // se	et the batch mode
cmbxSourceAlias.SetFocus; <	>procedure
	orm1.cmbxModeChange(Sender:
	bject);
nd. begi	<u> </u>
	cmbxMode.ItemIndex <> -1 then
	gin
if	
	tringsEqual(cmbxMode.Items[cm
if	Ī

bxMode.ItemIndex],'Append') then
BatchMove1.Mode := batAppend

<>object Label6: TLabel
Left = 344
Top = 48
Width = 25
Height = 13
Caption = 'Alias:'
end
<>object Label7: TLabel
Left = 324
Top = 88
Width = 61
Height = 13
Caption = 'Table Name:'
end
<>object Label8: TLabel
Left = 348
Top = 128
Width = 29
Height = 13
Caption = 'Index:'
end
<>object Bevel2: TBevel
Left = 16
Top = 8
Width = 265
Height = 169
Shape = bsFrame
Style = bsRaised
end
<>object Label5: TLabel
Left = 48
Top = 128
Width = 29
Height = 13
Caption = 'Index:'
end
<>object Label4: TLabel
Left = 24
Top = 88
Width = 61
Height = 13
Caption = 'Table Name:'
end
<>object Label3: TLabel
Left = 48
Top = 48
Width = 25
Height = 13

Objects36
Object <b>S</b>
الملف النصيّ لبرنامج
(BatchMv)
ا عامد ا
علاء الدين اللباد
واتف ۷۱۳۵۷۵۶۹۰ واتف ۱۷۳۵۷۵۶۹۰
<>object Form1: TForm1
Left = 173
Top = 120
Width = 766
Height = 584
Caption = 'TBatchMove Example'
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = False
Position = poDesktopCenter
OnCreate = FormCreate
OnShow = FormShow
PixelsPerInch = 96
TextHeight = 13
<>object Bevel1: TBevel
Left = 312
Top = 8
Width = 265
Height = 169
Shape = bsFrame
Style = bsRaised
end

Font.Height = -12
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
<>object Label9: TLabel
Left = 28
Top = 336
Width = 61
Height = 13
Caption = 'Batch Mode:'
end
<>object Label12: TLabel
Left = 288
Top = 232
Width = 110
Height = 13
Caption = 'Changed Table Name:
'
end
<>object Label13: TLabel
Left = 272
Top = 272
Width = 128
Height = 13
Caption = 'Key Violation Table
Name: '
end
<>object Label14: TLabel
Left = 296
Top = 312
Width = 105
Height = 13
Caption = 'Problem Table Name: '
end
<>object Label11: TLabel
Left = 256
Top = 352
Width = 145
Height = 13
Caption = 'Record Count (Leave 0
for all):'
end
<>object cmbxDestAlias:
TComboBox
Left = 400
Top = 48
Width = 145
1

Caption = 'Alias:'
end
<>object Bevel3: TBevel
Left = 16
Top = 192
Width = 561
Height = 201
Shape = bsFrame
Style = bsRaised
end
<>object Label1: TLabel
Left = 104
Top = 16
Width = 77
Height = 13
Caption = 'Source Table'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -12
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
<>object Label2: TLabel
Left = 400
Top = 16
Width = 101
Height = 13
Caption = 'Destination Table'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -12
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
<>object Label10: TLabel
Left = 224
Top = 200
Width = 131
Height = 13
Caption = 'Options/Error
Handling'
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
runt.Color — CI vy maow I ext

I a4 - 06
Left = 96
Top = 88 Width = 145
Height = 21
Style = csDropDownList
ItemHeight = 13
Sorted = True
TabOrder = 4
OnChange =
cmbxSourceTableChange
end
<>object cmbxSourceIndex:
TComboBox
Left = 96
Top = 128
Width = 145
Height = 21
Style = csDropDownList
ItemHeight = 13
Sorted = True
TabOrder = 5
OnChange =
cmbxSourceIndexChange
end
<>object cmbxMode:
TComboBox
Left = 96
Top = 336
Width = 145
Height = 21
Style = csDropDownList
ItemHeight = 13
Sorted = True
TabOrder = 6
OnChange = cmbxModeChange
Items.Strings = (
'Append'
'Append Update'
'Copy'
'Delete'
'Update')
end
<>object Button1: TButton
Left = 230
Top = 404
Width = 131
Height = 25
Caption = 'Execute BatchMove'

Height = 21
Style = csDropDownList
ItemHeight = 13
Sorted = True
TabOrder = 0
OnChange =
cmbxDestAliasChange
end
<>object cmbxDestTable:
TComboBox
Left = 400
Top = 88
Width = 145
Height = 21
ItemHeight = 13
Sorted = True
TabOrder = 1
OnChange =
cmbxDestTableChange
end
<>object cmbxDestIndex:
TComboBox
Left = 400
Top = 128
Width = 145
Height = 21
Style = csDropDownList
ItemHeight = 13 Sorted = True
TabOrder = 2
OnChange =
cmbxDestIndexChange
end
<>object cmbxSourceAlias:
TComboBox
Left = 96
Top = 48
Width = 145
Height = 21
Style = csDropDownList
ItemHeight = 13
Sorted = True
TabOrder = 3
OnChange =
cmbxSourceAliasChange
end
<>object cmbxSourceTable:
TComboBox

T. 1.O. 1. 44
TabOrder = 11
end
<>object edtKeyVioTbl:
TEdit
Left = 416
Top = 272
Width = 121
Height = 21
TabOrder = 12
end
<>object edtProbTbl: TEdit
Left = 416
Top = 312
Width = 121
Height = 21
TabOrder = 13
end
<>object edtRecCount: TEdit
Left = 416
Top = 352
Width = 121
Height = 21
TabOrder = 14
Text = '0'
OnKeyPress =
edtRecCountKeyPress
end
<>object BatchMove1:
TBatchMove
Destination = tblDest
Source = tblSource
Transliterate = False
Left = 280
Top = 88
end
<>object tblSource: TTable
Left = 280
Top = 24
end
<>object tblDest: TTable
Left = 280
Top = 144
end
End

TabOrder = 7
OnClick = Button1Click
end
<>object chkbxAbortKey:
TCheckBox
Left = 32
Top = 232
Width = 193
Height = 17
Caption = 'Abort on key violation'
Checked = True
State = cbChecked
TabOrder = 8
OnClick = chkbxAbortKeyClick
end
<>object
chkbxAbortProblem: TCheckBox
Left = 32
Top = 264
Width = 193
Height = 17
Caption = 'Abort on problem
(data is truncated)'
Checked = True
State = cbChecked
TabOrder = 9
OnClick =
chkbxAbortProblemClick
end
<>object chkbxTrans:
TCheckBox
Left = 32
Top = 296
Width = 193
Height = 17
Caption = 'Transliterate
Characters'
Checked = True
State = cbChecked
TabOrder = 10
OnClick = chkbxTransClick
end
<>object edtChangedTable:
TEdit
Left = 416
Top = 232
•
Width = 121
Height = 21

end
object Label7: TLabel
Left = 324
Top = 88
<b>Width = 61</b>
Height = 13
Caption = 'Table Name:'
end
object Label8: TLabel
Left = 348
Top = 128
Width = 29
Height = 13
Caption = 'Index:'
end
object Bevel2: TBevel
Left = 16
Top = 8
Width = 265
Height = 169
8
Shape = bsFrame
Style = bsRaised
end
object Label5: TLabel
Left = 48
Top = 128
Width = 29
Height = 13
Caption = 'Index:'
end
object Label4: TLabel
Left = 24
Top = 88
Width = 61
Height = 13
Caption = 'Table Name:'
end
object Label3: TLabel
Left = 48
Top = 48
Width = 25
Height = 13
Caption = 'Alias:'
end
object Bevel3: TBevel
Left = 16
Top = 192 Width = 561
Width = 561

ObjectS			
الملف النحي لبرنامج			
(BatchMv2)			
عامدا			
•			
علاء الدين اللباد			
واتف ۷۳۵۷۵۶۹۰			
object FormMain: TFormMain			
Left = 156			
Top = 110			
Width = 766			
Height = 584			
Caption = 'TBatchMove Example'			
Color = clBtnFace			
Font.Charset =			
DEFAULT CHARSET			
Font.Color = clWindowText			
Font.Height = -11 Font.Name = 'MS Sans Serif'			
Font.Style = []			
OldCreateOrder = False			
Position = poDesktopCenter			
OnCreate = FormCreate			
OnShow = FormShow			
PixelsPerInch = 96			
TextHeight = 13			
object Bevel1: TBevel			
Left = 312			
Top = 8			
Width = 265			
Height = 169			
Shape = bsFrame			
Style = bsRaised			
end			
object Label6: TLabel			
Left = 344			
Top = 48			
Width = 25			
Height = 13			
Caption = 'Alias:'			

Left = 28
Top = 336
Width = 61
Height = 13
Caption = 'Batch Mode:'
end
object Label12: TLabel
Left = 288
Top = 232
Width = 110
Height = 13
Caption = 'Changed Table Name:
end
object Label13: TLabel
Left = 272
Top = 272
Width = 128
Height = 13
Caption = 'Key Violation Table
Name: '
end
object Label14: TLabel
Left = 296
Top = 312
Width = 105
Height = 13
Caption = 'Problem Table Name: '
end
object Label11: TLabel
Left = 256
Top = 352
Width = 145
Height = 13
Caption = 'Record Count (Leave 0
for all):'
end
object cmbxDestAlias: TComboBox
Left = 400
Top = 48
Width = 145
Height = 21
Style = csDropDownList
ItemHeight = 13
Sorted = True
TabOrder = 0
OnChange =
cmbxDestAliasChange

Height = 201
Shape = bsFrame
Style = bsRaised
end
object Label1: TLabel
Left = 104
Top = 16
Width = 77
Height = 13
Caption = 'Source Table'
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -12
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
object Label2: TLabel
Left = 400
Top = 16
Width = 101
Height = 13
Caption = 'Destination Table'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -12
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
object Label10: TLabel
Left = 224
Top = 200
Width = 131
Height = 13
Caption = 'Options/Error
Handling'
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -12
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
object Label9: TLabel

TabOrder = 4
OnChange =
cmbxSourceTableChange
end
object cmbxSourceIndex:
TComboBox
Left = 96
Top = 128
Width = 145
Height = 21
Style = csDropDownList
ItemHeight = 13
Sorted = True
TabOrder = 5
OnChange =
cmbxSourceIndexChange
end
object cmbxMode: TComboBox
Left = 96
Top = 336
Width = 145
Height = 21
Style = csDropDownList
ItemHeight = 13
Sorted = True
TabOrder = 6
OnChange = cmbxModeChange
Items.Strings = (
'Append'
'Append Update'
'Copy'
'Delete'
'Update')
end
object Button1: TButton
Left = 230
Top = 404
Width = 131
Height = 25
Caption = 'Execute BatchMove'
TabOrder = 7
OnClick = Button1Click
end
object chkbxAbortKey:
TCheckBox
Left = 32
Top = 232
Width = 193

end
object cmbxDestTable:
TComboBox
Left = 400
Top = 88
Width = 145
Height = 21
ItemHeight = 13
Sorted = True
TabOrder = 1
OnChange =
cmbxDestTableChange
end
object cmbxDestIndex:
TComboBox
Left = 400
Top = 128
Width = 145
Height = 21
Style = csDropDownList
ItemHeight = 13
Sorted = True
TabOrder = 2
OnChange =
cmbxDestIndexChange
end
object cmbxSourceAlias:
TComboBox
Left = 96
Top = 48
<b>Width</b> = 145
Height = 21
Style = csDropDownList
ItemHeight = 13
Sorted = True
TabOrder = 3
OnChange =
S
cmbxSourceAliasChange
end
object cmbxSourceTable:
TComboBox
Left = 96
Top = 88
Width = 145
Height = 21
Style = csDropDownList
ItemHeight = 13
Sorted = True

Left = 416
Top = 312
Width = 121
Height = 24
TabOrder = 13
end
object edtRecCount: TEdit
Left = 416
Top = 352
Width = 121
Height = 24
TabOrder = 14
Text = '0'
OnKeyPress =
edtRecCountKeyPress
end
object BatchMove1: TBatchMove
Destination = tblDest
Source = tblSource
Transliterate = False
Left = 280
Top = 88
end
object tblSource: TTable
Left = 280
Top = 24
end
object tblDest: TTable
Left = 280
Top = 144
end
End

ObjectS الملف التنفيذي		
لبرنامج		
	(BatchMv2)	
عداد		
علاء الدين اللباد		
واتف ۹٤٤٥٧٥٣٧١		
unit Main;		

Height = 17
Caption = 'Abort on key violation'
Checked = True
State = cbChecked
TabOrder = 8
OnClick = chkbxAbortKeyClick
end
object chkbxAbortProblem:
TCheckBox
Left = 32
Top = 264
Width = 193
Height = 17
Caption = 'Abort on problem
(data is truncated)'
Checked = True
State = cbChecked
TabOrder = 9
OnClick =
chkbxAbortProblemClick
end
object chkbxTrans: TCheckBox
Left = 32
Top = 296
Width = 193
Height = 17
Caption = 'Transliterate
Characters'
Checked = True
State = cbChecked
TabOrder = 10
OnClick = chkbxTransClick
end
object edtChangedTable: TEdit
Left = 416
Top = 232
Width = 121
Height = 24
TabOrder = 11
end
object edtKeyVioTbl: TEdit
Left = 416
Top = 272
Width = 121
Height = 24
TabOrder = 12
end
object edtProbTbl: TEdit
o-jest sationini inait

procedure
cmbxSourceAliasChange(Sender:
TObject);
procedure
cmbxDestAliasChange(Sender:
TObject);
procedure
cmbxSourceTableChange(Sender:
TObject);
procedure
cmbxDestTableChange(Sender:
TObject);
procedure
cmbxSourceIndexChange(Sender:
TObject);
procedure
•
cmbxDestIndexChange(Sender:
TObject);
procedure Button1Click(Sender:
TObject);
procedure
chkbxAbortKeyClick(Sender:
TObject);
procedure
chkbxAbortProblemClick(Sender:
TObject);
procedure
<pre>chkbxTransClick(Sender: TObject);</pre>
procedure
cmbxModeChange(Sender:
TObject);
procedure
edtRecCountKeyPress(Sender:
TObject; var Key: Char);
procedure FormShow(Sender:
TObject);
private
{ Private declarations }
function IsStringsEqual(const
s1,s2 : string): boolean;
public
{ Public declarations }
end;
var
FormMain: TFormMain;
implementation

interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
StdCtrls, ExtCtrls, Db, DBTables,
ComCtrls;
TEATH Main = alogg(TEATH)
TFormMain = class(TForm)  BatchMove1: TBatchMove;
tblSource: TTable;
,
tblDest: TTable;
Bevel1: TBevel;
Label6: TLabel;
cmbxDestAlias: TComboBox;
Label7: TLabel;
cmbxDestTable: TComboBox;
Label8: TLabel;
cmbxDestIndex: TComboBox;
Bevel2: TBevel;
cmbxSourceAlias: TComboBox;
cmbxSourceTable: TComboBox;
cmbxSourceIndex: TComboBox;
Label5: TLabel;
Label4: TLabel;
Label3: TLabel;
cmbxMode: TComboBox;
Button1: TButton;
Bevel3: TBevel;
chkbxAbortKey: TCheckBox;
chkbxAbortProblem: TCheckBox;
chkbxTrans: TCheckBox;
Label1: TLabel;
Label2: TLabel;
Label10: TLabel;
Label9: TLabel;
Label12: TLabel;
edtChangedTable: TEdit;
Label13: TLabel;
edtKeyVioTbl: TEdit;
Label14: TLabel;
edtProbTbl: TEdit;
Label11: TLabel;
edtRecCount: TEdit;
procedure FormCreate(Sender:
TObject);

begin	{\$R *.DFM}
tblDest.DatabaseName :=	
cmbxDestAlias.Items[cmbxDestAlias	resourcestring
.ItemIndex];	sIncomplete = 'Incomplete input.';
	sCompleted = 'BatchMove
Session.GetTableNames(tblDest.Dat	complete. Number of records
abaseName,'',true,false,cmbxDestTa	applied: ';
ble.Items);	upparent ,
end	procedure
else	TFormMain.FormCreate(Sender:
begin	TObject); // Get alias names
tblDest.DatabaseName := ";	begin
cmbxDestTable.Items.Clear;	if Session.Active = false then
end;	Session.Open;
end;	
	Session.GetAliasNames(CmbxSourc
procedure	eAlias.Items);
TFormMain.cmbxSourceTableChan	cmbxDestAlias.Items :=
ge(Sender: TObject);	cmbxSourceAlias.Items;
begin	end;
if cmbxSourceTable.ItemIndex <> -	
1 then	procedure
begin	TFormMain.cmbxSourceAliasChan
tblSource.TableName :=	ge(Sender: TObject);
cmbxSourceTable.Items[cmbxSourc	begin
eTable.ItemIndex];	if cmbxSourceAlias.ItemIndex <> -
,	1 then
tblSource.GetIndexNames(cmbxSou	begin
rceIndex.Items);	tblSource.DatabaseName :=
end	cmbxSourceAlias.Items[cmbxSource
else	Alias.ItemIndex];
	Alias.ttellifidex[,
begin	Session CotTableNews(thlSeynes D
tblSource.TableName := '';	Session.GetTableNames(tblSource.D
cmbxSourceIndex.Items.Clear;	atabaseName,",true,false,cmbxSour
end;	ceTable.Items);
	end
end;	else
	begin
procedure	tblSource.DatabaseName := '';
TFormMain.cmbxDestTableChange	cmbxSourceTable.Items.Clear;
(Sender: TObject);	end;
begin	end;
if cmbxDestTable.ItemIndex <> -1	
then	procedure
begin	TFormMain.cmbxDestAliasChange(
tblDest.TableName :=	Sender: TObject);
cmbxDestTable.Items[cmbxDestTab	begin
le.ItemIndex];	if cmbxDestAlias.ItemIndex <> -1
re-reminues,	then
	UICII

10 (1175 - (75 11 3) - 11 (1	
if tblDest.TableName = " then	tblDest.GetIndexNames(cmbxDestIn
tblDest.TableName :=	dex.Items);
cmbxDestTable.Text;	end
if ((tblSource.DatabaseName <> '')	else
and // test for enough input	begin
(tblSource.TableName <> '') and	tblDest.TableName := '';
(tblDest.DatabaseName <> '') and	cmbxDestIndex.Items.Clear;
(tblDest.TableName <> '') and	end;
	end;
(cmbxMode.items[cmbxMode.ItemI	,
ndex  <> '')) then	procedure
begin	TFormMain.cmbxSourceIndexChan
BatchMove1.ChangedTableName	ge(Sender: TObject);
:= edtChangedTable.Text; // more	begin
batchmove setup	if cmbxSourceIndex.ItemIndex <> -
BatchMove1.KeyViolTableName	1 then
:= edtKeyVioTbl.Text;	begin
BatchMove1.ProblemTableName	tblSource.IndexName :=
:= edtProbTbl.Text;	
BatchMove1.RecordCount :=	cmbxSourceIndex.Items[cmbxSourc
	eIndex.ItemIndex];
StrToInt(edtRecCount.Text);	end
end	else
else	begin
begin	tblSource.IndexName := '';
MessageDlg(sIncomplete,	end;
mtError, [mbOK], 0);	end;
exit;	
end;	procedure
BatchMove1.Execute; // run the	TFormMain.cmbxDestIndexChange
batchmove	(Sender: TObject);
MessageDlg(sCompleted +	begin
IntToStr(BatchMove1.MovedCount)	if cmbxDestIndex.ItemIndex <> -1
,mtInformation,[mbOK],0);	then
end;	begin
	tblDest.IndexName :=
procedure	cmbxDestIndex.Items[cmbxDestInde
TFormMain.chkbxAbortKeyClick(S	x.ItemIndex];
ender: TObject);	end
begin	else
BatchMove1.AbortOnKeyViol :=	begin
chkbxAbortKey.Checked;	
end;	tblDest.IndexName := ";
Ciru,	end;
procedure	
procedure TFormMain.chkbxAbortProblemCli	end;
ck(Sender: TObject);	procedure
begin	TFormMain.Button1Click(Sender:
BatchMove1.AbortOnProblem :=	TObject);
chkbxAbortProblem.Checked;	begin

MessageDlg('Batch mode not	
found',mtError,[mbOK],0);	
end;	
end;	
,	
// only allow numbers to be typed i	in
procedure	
TFormMain.edtRecCountKeyPres	ss(
Sender: TObject; var Key: Char);	· 
begin	
if ((key in ['0''9'] = false) and	
(word(key) $\Leftrightarrow$ VK BACK)) then	
key := #0;	
end;	
procedure	
TFormMain.FormShow(Sender:	
TObject);	
begin	
cmbxSourceAlias.SetFocus;	
end;	
/	
end.	

end;
procedure
TFormMain.chkbxTransClick(Send
er: TObject);
begin
BatchMove1.Transliterate :=
chkbxTrans.Checked;
end;
function
TFormMain.IsStringsEqual(const
s1,s2 : string): boolean;
begin
Result := UpperCase(s1) =
UpperCase(s2);
end;
// set the batch mode
procedure
TFormMain.cmbxModeChange(Sen
der: TObject);
begin
if cmbxMode.ItemIndex <> -1 then
begin
if
IsStringsEqual(cmbxMode.Items[cm
bxMode.ItemIndex],'Append') then
BatchMove1.Mode := batAppend
else if
IsStringsEqual(cmbxMode.Items[cm
bxMode.ItemIndex],'Copy') then
BatchMove1.Mode := batCopy
else if
IsStringsEqual(cmbxMode.Items[cm
bxMode.ItemIndex],'Append
Update') then
BatchMove1.Mode :=
<b>batAppendUpdate</b>
else if
IsStringsEqual(cmbxMode.Items[cm
bxMode.ItemIndex],'Delete') then
BatchMove1.Mode := batDelete
else if
IsStringsEqual(cmbxMode.Items[cm
bxMode.ItemIndex],'Update') then
BatchMove1.Mode := batUpdate
else

// the scrolling behavior. no attempt
is made to verify that the parameters
// supplied by the user make any
sense.
//
unit scroldem;
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
Menus, StdCtrls, ComCtrls,
ExtCtrls;
type
TScrollForm1 = class(TForm)
Bevel1: TBevel;
,
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Label1: TLabel;
Vertical: TLabel;
ImgPanel: TPanel;
Image: TImage;
HScrollb: TScrollBar;
VScrollb: TScrollBar;
HUnits: TEdit;
HPage: TEdit;
HMax: TEdit;
VUnits: TEdit;
VPage: TEdit;
VMax: TEdit;
HUnitsUpDown: TUpDown;
HPageUpDown: TUpDown;
HMaxUpDown: TUpDown;
VUnitsUpDown: TUpDown;
VMaxUpDown: TUpDown;
VPageUpDown: TUpDown;
DefaultBtn: TButton;
ApplyBtn: TButton;
MainMenu1: TMainMenu;
File1: TMenuItem;
<u>,                                      </u>
Open1: TMenuItem;
N1: TMenuItem;
Exit1: TMenuItem;
View1: TMenuItem;

Object8
الملف التنفيذي لبرنامج
(Scroll bar)
إعداد
علاء الدين اللباد
واتف ۷۱۳۵۷۵۶۹۰
الملف التنفيذي للبرنامج
program scroll;
وفيه
Procedures 38
program scroll;
uses
Forms,
scroldem in 'scroldem.pas'
{ScrollForm1},
about in 'about.pas' {AboutBox};
(6D + DEC)
{\$R *.RES}
hagin
begin Application.Initialize;
Application.imuanze;
Application.CreateForm(TScrollForm1, ScrollForm1);
Application.Run;
end
Ciiu.
//
// this example demonstrates the use
of TScrollBar, including how to
// properly set scrolling ranges and
how to respond to the scroll bar
// notification events. in addition, an
interface is provided to the
// user to adjust the values to see how
the different parameters affect

UpdateDisplay; property Dirty: Boolean read FDirty write SetDirty;  Dublic  //constructor Create(Owner: TComponent); override; { Public declarations } end;  var  ScrollForm1: TScrollForm1;  ScrollForm1: TScrollForm1;  Implementation  ScrollForm1: TScrollForm1;  ScrollForm1: TScrollForm1;  Const  DEF_SCROLL_UNITS = 8;  Const  DEF_SCROLL_UNITS = 8;  Const  TObject); begin ScrollReset; end;  Const  Const  DEF_SCROLL_UNITS = 8;  Const  ScrollForm1.FormCreate(Sender: TObject); begin ScrollForm1.Exit1Click(Sender: TObject); begin Close; end;  Close; end;  HScrollForm1.View1Click(Sender: TObject); begin  HScrollBorm1.View1Click(Sender: TObject); begin  HScrollBorm1.View1Click(Sender: TObject); begin  HScrollMenu.Checked := HScrollb.Visible; VScrollMenu.Checked := HScrollb.Visible; end;		
property Dirty: Boolean read FDirty write SetDirty;  Dublic  //constructor Create(Owner: TComponent); override; { Public declarations } end;  var  ScrollForm1: TScrollForm1;  implementation  HSc Uses about;  Const  DEF_SCROLL_UNITS = 8;  Const  DEF_SCROLL_UNITS = 8;  Const  DEF_SCROLL_UNITS = 8;  Const  TObject); begin ScrollReset; end;  Close; end;  Close; end;  HSc Close; end;  HSc Close; end;  FScrollForm1.View1Click(Sender: TObject); begin HScrollMenu.Checked := HSc Core  ScrollReset  Core  ScrollReset	-	V
public //constructor Create(Owner: TComponent); override; { Public declarations } end;  var ScrollForm1: TScrollForm1;  implementation  uses about;  const DEF_SCROLL_UNITS = 8;  const Const Const DEF_SCROLL_UNITS = 8;  const Con		H
public //constructor Create(Owner: TComponent); override; { Public declarations } end;  var  ScrollForm1: TScrollForm1;  implementation  uses about;  const  DEF_SCROLL_UNITS = 8;  const  DEF_SCROLL_UNITS = 8;  const  DEF_SCROLL_UNITS = 8;  const  TObject); begin  ScrollReset; end; > procedure TScrollForm1.Exit1Click(Sender: TObject); begin  Close; end; > procedure TScrollForm1.View1Click(Sender: TObject); begin	property Dirty: Boolean read	Н
public //constructor Create(Owner: TComponent); override; { Public declarations } end;  var  ScrollForm1: TScrollForm1;  implementation  uses about;  const  DEF_SCROLL UNITS = 8;  const  DEF_SCROLL UNITS = 8;  const  TScrollForm1.FormCreate(Sender: TObject); begin ScrollReset; end; > procedure TScrollForm1.Exit1Click(Sender: TObject); begin Close; end; > procedure TScrollForm1.View1Click(Sender: TObject); begin HScrollMenu.Checked := HScrollb.Visible; VScrollMenu.Checked := UScrollB.Visible; VScrollB.Visible;	FDirty write SetDirty;	Α
public //constructor Create(Owner: TComponent); override; { Public declarations } end;  var  ScrollForm1: TScrollForm1;  implementation  uses about;  const  DEF_SCROLL UNITS = 8;  const  DEF_SCROLL UNITS = 8;  const  TScrollForm1.FormCreate(Sender: TObject); begin ScrollReset; end; > procedure TScrollForm1.Exit1Click(Sender: TObject); begin Close; end; > procedure TScrollForm1.View1Click(Sender: TObject); begin HScrollMenu.Checked := HScrollb.Visible; VScrollMenu.Checked := UScrollB.Visible; VScrollB.Visible;		О
TComponent); override; { Public declarations } end;  var  ScrollForm1: TScrollForm1;  implementation  Uses about;  const  DEF_SCROLL_UNITS = 8;  const  DEF_SCROLL_UNITS = 8;  const  TObject); begin  ScrollReset; end;  Close; end;  Close; end;  HScrollForm1.Exit1Click(Sender: TObject); begin  Close; end;  Close; end;  HScrollForm1.View1Click(Sender: TObject); begin  HScrollMenu.Checked := HScrollBenu.Checked := HScrollBelia in Close; end;  Close; end; Exital Close; Exital Close; Exital Close; Ex	public	<.
TComponent); override; { Public declarations } end;  var  ScrollForm1: TScrollForm1;  implementation  Uses about;  const  DEF_SCROLL_UNITS = 8;  const  DEF_SCROLL_UNITS = 8;  const  TObject); begin  ScrollReset; end;  Close; end;  Close; end;  HScrollForm1.Exit1Click(Sender: TObject); begin  Close; end;  Close; end;  HScrollForm1.View1Click(Sender: TObject); begin  HScrollMenu.Checked := HScrollBenu.Checked := HScrollBelia in Close; end;  Close; end; Exital Close; Exital Close; Exital Close; Ex	//constructor Create(Owner:	For
Exit	`	<.
end;  var  ScrollForm1: TScrollForm1;  implementation  uses about;  Const  DEF SCROLL UNITS = 8;  Const  DEF SCROLL UNITS = 8;  Corost  TObject);  begin  ScrollReset; end;  Close; end;  Close; end;  HScrollForm1.View1Click(Sender: TObject); begin  HScrollMenu.Checked := HScrollb.Visible; VScrollMenu.Checked := VScrollBe; VScrollMenu.Checked := VScrollBe; VScrollMenu.Checked := Uien  View VScrollMenu.Checked := VScrollMenu.Checked := VScrollMenu.Checked := VScrollMenu.Checked := VScrollMenu.Checked := VScrollBe; VScrollMenu.Checked := VScrollMenu		Exit
Vier ScrollForm1: TScrollForm1;  implementation  uses about;  const  DEF SCROLL UNITS = 8;  const  TObject); begin ScrollReset; end; > procedure TScrollForm1.Exit1Click(Sender: TObject); begin Close; end; > procedure TScrollForm1.Exit1Click(Sender: TObject); begin Close; end; > procedure TScrollForm1.Exit1Click(Sender: TObject); begin Close; end; > procedure TScrollForm1.View1Click(Sender: TObject); begin> procedure TScrollMenu.Checked := HScrollb.Visible; VScrollMenu.Checked := UScrollb.Visible;		<
ScrollForm1: TScrollForm1;  implementation  uses about;  (SR *.DFM)  Const  DEF SCROLL UNITS = 8;  C———> procedure TScrollForm1.FormCreate(Sender: TObject); begin ScrollReset; end;  C———> procedure TScrollForm1.Exit1Click(Sender: TObject); begin Close; end;  Close; end; end;  Close; end; en	•114)	<b>-</b>
ScrollForm1: TScrollForm1;  implementation  uses about;  Const  DEF SCROLL UNITS = 8;  Const  TScrollForm1.FormCreate(Sender: TObject); begin  ScrollReset; end;  Close; end;  Close; end;  TScrollForm1.Exit1Click(Sender: TObject); begin  Close; end;  Close; end;  HScrollForm1.View1Click(Sender: TObject); begin  HScrollMenu.Checked := HScrollb.Visible;  VScrollMenu.Checked := UScrollb.Visible; end;	var	1
implementation  uses about;  (SR *.DFM}  const  DEF_SCROLL_UNITS = 8;  const  TObject);  begin  ScrollReset; end;  Close; end;  Close; end;  Close; end;  TScrollForm1.View1Click(Sender: TObject); begin  HScrollMenu.Checked := HScrollMenu.Checked := UScrollMenu.Checked := USc		
implementation  uses about;  Cope  {\$R*.DFM}  Const  DEF_SCROLL_UNITS = 8;  C> procedure TScrollForm1.FormCreate(Sender: TObject); begin ScrollReset; end;  Close; end;	Scront of mr. 1Scront of mr,	<b>-</b>
uses about;  (SR *.DFM)  const  DEF_SCROLL_UNITS = 8;  Const  Seroul Scrooll Scrool Scrooll Scrool Scrool Scrool Scrooll Scrool Scroo	implementation	-
uses about;    SR *.DFM    Scoots	пприетентация	_
SR *.DFM   VSc   Const	1 4	<b>-</b>
{\$R *.DFM}  const  DEF_SCROLL_UNITS = 8;  <> procedure TScrollForm1.FormCreate(Sender: TObject); begin ScrollReset; end; > procedure TScrollForm1.Exit1Click(Sender: TObject); begin Close; end; > procedure TScrollForm1.View1Click(Sender: TObject); begin > procedure> procedure> procedure	uses about;	_
<pre> const  DEF_SCROLL_UNITS = 8;  &lt;&gt; procedure TScrollForm1.FormCreate(Sender: TObject); begin ScrollReset; end;  &lt;&gt; procedure TScrollForm1.Exit1Click(Sender: TObject); begin Close; end; &gt; procedure TScrollForm1.View1Click(Sender: TObject); begin Close; end; &gt; procedure TScrollForm1.View1Click(Sender: TObject); begin HScrollMenu.Checked := HScrollb.Visible; VScrollMenu.Checked := UScrollb.Visible; end; </pre>		
const  DEF_SCROLL_UNITS = 8;  <> procedure TScrollForm1.FormCreate(Sender: TObject); begin ScrollReset; end;  <> procedure TScrollForm1.Exit1Click(Sender: TObject); begin Close; end; > procedure TScrollForm1.View1Click(Sender: TObject); begin  Close; end; > procedure TScrollForm1.View1Click(Sender: TObject); begin		
const  DEF_SCROLL_UNITS = 8; > procedure TScrollForm1.FormCreate(Sender: TObject); begin ScrollReset; end; > procedure TScrollForm1.Exit1Click(Sender: TObject); begin Close; end; > procedure TScrollForm1.View1Click(Sender: TObject); begin HScrollMenu.Checked := HScrollb.Visible; VScrollMenu.Checked := VScrollb.Visible; end;	{\$R *.DFM}	
DEF_SCROLL_UNITS = 8;  <> procedure TScrollForm1.FormCreate(Sender: TObject); begin ScrollReset; end;  <> procedure TScrollForm1.Exit1Click(Sender: TObject); begin Close; end; > procedure TScrollForm1.View1Click(Sender: TObject); begin  HScrollForm1.View1Click(Sender: TObject); begin > procedure TScrollForm1.View1Click(Sender: TObject); begin > procedure TScrollForm1.View1Click(Sender: TObject); begin > procedure TScrollForm1.View1Click(Sender: TObject); begin> procedure TScrollForm1.View1Click(Sender: TObject); begin		
<pre> &lt;&gt; procedure TScrollForm1.FormCreate(Sender: TObject); begin ScrollReset; end;  &lt;&gt; procedure TScrollForm1.Exit1Click(Sender: TObject); begin Close; end;  &lt;&gt; procedure TScrollForm1.View1Click(Sender: TObject); begin  &lt;&gt; procedure TScrollForm1.View1Click(Sender: TObject); begin &gt; procedure TScrollForm1.View1Click(Sender: TObject); begin&gt; procedure TScrollForm1.View1Click(Sender: TObject); begin</pre>	const	
<pre>&lt;&gt; procedure TScrollForm1.FormCreate(Sender: TObject); begin ScrollReset; end; &lt;&gt; procedure TScrollForm1.Exit1Click(Sender: TObject); begin Close; end;  &lt;&gt; procedure TScrollForm1.View1Click(Sender: TObject); begin </pre> <pre></pre>	<pre>DEF_SCROLL_UNITS = 8;</pre>	
TScrollForm1.FormCreate(Sender: TObject); begin ScrollReset; end;  <> procedure TScrollForm1.Exit1Click(Sender: TObject); begin Close; end; > procedure TScrollForm1.View1Click(Sender: TObject); begin > procedure TScrollForm1.View1Click(Sender: TObject); begin> procedure TScrollForm1.View1Click(Sender: TObject); begin> procedure TScrollForm1.View1Click(Sender: TObject); begin> procedure TScrollForm1.View1Click(Sender: TObject); begin> procedure TScrollForm1.View1Click(Sender: TObject); begin		
TScrollForm1.FormCreate(Sender: TObject); begin ScrollReset; end; > procedure TScrollForm1.Exit1Click(Sender: TObject); begin Close; end; > procedure TScrollForm1.View1Click(Sender: TObject); begin > procedure TScrollForm1.View1Click(Sender: TObject); begin> procedure TScrollForm1.View1Click(Sender: TObject); begin> procedure TScrollForm1.View1Click(Sender: TObject); begin> procedure> procedure> procedure> procedure	<> procedure	
TObject); begin ScrollReset; end;  <> procedure TScrollForm1.Exit1Click(Sender: TObject); begin Close; end; > procedure TScrollForm1.View1Click(Sender: TObject); begin > procedure TScrollForm1.View1Click(Sender: TObject); begin HScrollMenu.Checked := HScrollb.Visible; VScrollMenu.Checked := VScrollb.Visible; end;		<-
begin ScrollReset; end; > procedure TScrollForm1.Exit1Click(Sender: TObject); begin Close; end; > procedure TScrollForm1.View1Click(Sender: TObject); begin > procedure TScrollForm1.View1Click(Sender: TObject); begin> begin> procedure TScrollForm1.View1Click(Sender: TObject); begin> procedure TScrollBorm1.View1Click(Sender: TObject); begin> procedure		Img
ScrollReset; end;  Abo > procedure TScrollForm1.Exit1Click(Sender: TObject); begin Close; end; > procedure TScrollForm1.View1Click(Sender: TObject); begin  HScrollMenu.Checked:= HScrollb.Visible; VScrollb.Visible; end;	• / /	<b>    &lt;</b> .
end;  <> procedure TScrollForm1.Exit1Click(Sender: TObject); begin Close; end; > procedure TScrollForm1.View1Click(Sender: TObject); begin HScrollMenu.Checked := HScrollb.Visible; VScrollMenu.Checked := VScrollb.Visible; end;		Abo
<pre></pre>	,	†   <.
TScrollForm1.Exit1Click(Sender: TObject); begin Close; end; > procedure TScrollForm1.View1Click(Sender: TObject); begin HScrollMenu.Checked:= HScrollb.Visible; VScrollMenu.Checked:= VScrollb.Visible; end;	- Circle	App
TScrollForm1.Exit1Click(Sender: TObject); begin Close; end; > procedure TScrollForm1.View1Click(Sender: TObject); begin HScrollMenu.Checked := HScrollb.Visible; VScrollb.Visible; end;	<> nrocedure	- <
TObject); begin Close; end;  <> procedure TScrollForm1.View1Click(Sender: TObject); begin HScrollMenu.Checked := HScrollb.Visible; VScrollMenu.Checked := VScrollb.Visible; end;  <	<u> </u>	Defa
begin Close; end; > procedure TScrollForm1.View1Click(Sender: TObject); begin HScrollMenu.Checked := HScrollb.Visible; VScrollMenu.Checked := VScrollb.Visible; end;	`	<-
Close; end; > procedure TScrollForm1.View1Click(Sender: TObject); begin HScrollMenu.Checked := HScrollb.Visible; VScrollMenu.Checked := VScrollb.Visible; end;	<b>V</b> //	UpI
end;  <> procedure  TScrollForm1.View1Click(Sender: TObject);  begin  HScrollMenu.Checked := HScrollb.Visible;  VScrollMenu.Checked := VScrollb.Visible; end;  end;	C	But
<pre>c&gt; procedure TScrollForm1.View1Click(Sender: TObject); begin HScrollMenu.Checked := HScrollb.Visible; VScrollMenu.Checked := VScrollb.Visible; end;</pre> <pre>c</pre> Ima Scroll Scr	,	
<pre> &lt;&gt; procedure TScrollForm1.View1Click(Sender: TObject); begin HScrollMenu.Checked := HScrollb.Visible; VScrollMenu.Checked := VScrollb.Visible; end;  {</pre>	ena;	nri
TScrollForm1.View1Click(Sender: TObject); begin HScrollMenu.Checked := HScrollb.Visible; VScrollMenu.Checked := VScrollb.Visible; end; U  I  U  I  I  I  I  I  I  I  I  I  I		
TObject); begin HScrollMenu.Checked := HScrollb.Visible; VScrollMenu.Checked := VScrollb.Visible; end;  F  Image: Scroll of the content of th	<u> </u>	
begin  HScrollMenu.Checked :=  HScrollb.Visible;  VScrollMenu.Checked :=  VScrollb.Visible;  end;  Scro	· ·	l —
HScrollMenu.Checked := HScrollb.Visible; VScrollMenu.Checked := VScrollb.Visible; end;  Scro	<b>V</b> //	
HScrollMenu.Checked := HScrollb.Visible; VScrollMenu.Checked := VScrollb.Visible; end;  Scro		<u> </u>
VScrollMenu.Checked := VScrollb.Visible; end;  Scro		
VScrollMenu.Checked :=  VScrollb.Visible; end;  Ima  <		
end; Scro	VScrollMenu.Checked :=	
end; Scro	VScrollb.Visible;	
	end;	]   <-
<> procedure		Scro
	<> procedure	<.

ATO HAY TONK I
VScrollMenu: TMenuItem;
HScrollMenu: TMenuItem;
Help1: TMenuItem;
About1: TMenuItem;
OpenDialog: TOpenDialog;
<> procedure
FormCreate(Sender: TObject);
<> procedure
Exit1Click(Sender: TObject);
<> procedure
View1Click(Sender: TObject);
<> procedure
VScrollMenuClick(Sender:
TObject);
<> procedure
HScrollMenuClick(Sender:
TObject);
<> procedure
Open1Click(Sender: TObject);
<> procedure
VScrollbChange(Sender: TObject);
<> procedure
HScrollbScroll(Sender: TObject;
ScrollCode: TScrollCode; var
ScrollPos: Integer);
<> procedure
ImgPanelResize(Sender: TObject);
<> procedure
About1Click(Sender: TObject);
<> procedure
ApplyBtnClick(Sender: TObject);
<> procedure
DefaultBtnClick(Sender: TObject);
<> procedure
UpDownClick(Sender: TObject;
Button: TUDBtnType);
private
{ Private declarations }
Units: TPoint;
FDirty: Boolean;
T DII ty. Duulcan,
2 S 1 C /D' / /1
<> procedure SetDirty(d:
Boolean);
<> procedure
ImageLoad(filename: string);
<> procedure
ScrollAdjust(update: Boolean);
<> procedure ScrollReset;
r- ottaart zeromiteset)

needed for a few reasons:	TScrollForm1.VScrollMenuClick(Se
// 1) you want to do something	nder: TObject);
special with particular scroll	begin
// events	VScrollb.Visible := not
// 2) the range of your scrollbar is	VScrollb.Visible;
larger than [0, 65535]. in	end;
// this case, the ScrollPos element	Ź
is invalid and much of the	<> procedure
// scrolling has to be done	TScrollForm1.HScrollMenuClick(Se
manually.	nder: TObject);
	begin
<> procedure	HScrollb.Visible := not
TScrollForm1.HScrollbScroll(Sende	HScrollb. Visible;
r: TObject; ScrollCode:	
1	end;
TScrollCode; var ScrollPos:	
Integer);	<> procedure
begin	TScrollForm1.Open1Click(Sender:
if HScrollb.Max <= 65535 then	TObject);
Image.Left := -Units.x * ScrollPos	begin
else	if OpenDialog.Execute then
ShowMessage('Scrollbar ranges >	
65535 not supported');	ImageLoad(OpenDialog.FileName);
end;	end;
//	//
// The panel's size doesn't change in	// Hooking the OnChange event is
this example, so	the simplest way to handle the
// this event will only happen once.	// scrollbar notification. Note the
In general, the range	check to ensure that the
// of the scrollbars will need to be	// that the sender is the vertical
changed when the size of	scrollbar. While irrelevant for
// the visible window is changed.	// this example, some forms may
// the visible window is changed.	
	need to distinguish between various
<> procedure	// scrollbars.
TScrollForm1.ImgPanelResize(Send	//
er: TObject);	<> procedure
begin	TScrollForm1.VScrollbChange(Send
ScrollAdjust(True);	er: TObject);
end;	begin
	if Sender as TScrollBar = VScrollb
function scale(n, u: Integer):	then
Integer;	Image.Top := -Units.y *
begin	VScrollb.Position;
<b>Result</b> := (n + u - 1) div u;	end;
end;	
,	//
function maxval(a, b: Integer):	// Hooking the OnScroll event is
Integer;	another option for handling the
	// scrollbar notification. This is
begin	// scrombar nouncation. This is

//is ignored	if a >= b then
// is ignored.	Result := a
<> procedure TScrollForm1.ScrollReset;	else Bosult := b:
	Result := b;
begin	end;
// this fixed size works nice,	11
although in some cases basing	//
// the scroll units upon the size of	// Readjust the range of the
the image works well, too.	scrollbars to match the new size
Units.x := DEF_SCROLL_UNITS;	// of the visible window. (Also called
Units.y := DEF_SCROLL_UNITS;	by ScrollReset to avoid
101 Di di ni 11 di	// duplicating the code).
if Image.Picture <> nil then	//
begin	<> procedure
// move the image and the	TScrollForm1.ScrollAdjust(update:
scrollbars to their "home" location	Boolean);
Image.Top := 0;	begin
Image.Left := 0;	// set the maximum scrolling to be
HScrollb.Position := 0;	large enough to scroll the whole
VScrollb.Position := 0;	// image, but not so much that it can
	scroll off the window. we
// negative scrolling ranges aren't	// divide by the number of Units
worth the trouble, so set	(>1) because scrolling one pixel at
// the range to go from [0, M]. See	// a time is way too slow for most
'ScrollAdjust' to see how M	images of any size.
// is calculated.	
HScrollb.Min := 0;	if Image.Picture ⇔ nil then
VScrollb.Min := 0;	begin
ScrollAdjust(False);	HScrollb.Max := maxval(0,
	scale(Image.Width -
HScrollb.LargeChange :=	ImgPanel.Width, Units.x));
scale(HScrollb.Max, Units.x);	VScrollb.Max := maxval(0,
VScrollb.LargeChange :=	scale(Image.Height -
scale(VScrollb.Max, Units.y);	ImgPanel.Height, Units.y));
	end;
HScrollb.Visible := True;	
VScrollb.Visible := True;	if update then
end;	UpdateDisplay;
	end;
UpdateDisplay;	
end;	//
,	// ScrollReset() is used to put the
	scrollbars into a sane "initial"
<> procedure	// state. this needs to be done each
TScrollForm1.SetDirty(d: Boolean);	time the image is changed,
begin	// since the amount we scroll depends
if d <> FDirty then	upon the image size. for
begin	// simplicity, the special case of "new
~~SIII	image size" == "old image size"
	Se vine

	FDirty := d;
<> procedure	ApplyBtn.Enabled := FDirty;
TScrollForm1.DefaultBtnClick(Send	end;
er: TObject);	end;
begin	
ScrollReset;	
Dirty := False;	<> procedure
end;	TScrollForm1.ImageLoad(filename:
	string);
<> procedure	begin
TScrollForm1.UpDownClick(Sender	
: TObject; Button: TUDBtnType);	Image.Picture.LoadFromFile(filena
begin	me);
Dirty := True;	ScrollReset;
end;	end;
- Chu,	chu,
<> procedure	<> procedure
TScrollForm1.UpdateDisplay;	TScrollForm1.About1Click(Sender:
begin	TObject);
HUnitsUpDown.Position := Units.x;	var
-	about: TAboutBox;
HPageUpDown.Position :=	
HScrollb.LargeChange;	begin
HMaxUpDown.Position :=	about := TAboutBox.Create(Self);
HScrollb.Max;	try
NITT OF THE PARTY	about.ShowModal;
VUnitsUpDown.Position := Units.y;	finally
VPageUpDown.Position :=	about.Free;
VScrollb.LargeChange;	end;
VMaxUpDown.Position :=	end;
VScrollb.Max;	
Dirty := False;	<> procedure
end;	TScrollForm1.ApplyBtnClick(Sende
	r: TObject);
end.	begin
	<b>Units.x</b> := <b>HUnitsUpDown.Position</b> ;
	HScrollb.LargeChange :=
	HPageUpDown.Position;
unit about;	HScrollb.Max :=
	HMaxUpDown.Position;
interface	-
	Units.y := VUnitsUpDown.Position;
uses	VScrollb.LargeChange :=
Windows, Messages, SysUtils,	VPageUpDown.Position;
Classes, Graphics, Controls, Forms,	VScrollb.Max :=
Dialogs,	VMaxUpDown.Position;
StdCtrls;	. Manage of the state of the st
> \( \tag{1.15}	Dirty := False;
tyne	end;
type	ciiu,

biMinimize]
BorderStyle = bsSingle
Caption = 'ScrollDemo'
ClientHeight = 194
ClientWidth = 436
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
Menu = MainMenu1
OldCreateOrder = True
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Bevel1: TBevel
Left = 0
Top = 0
Width = 436
Height = 9
Align = alTop
Shape = bsTopLine
end
<> object Label3: TLabel
Left = 238
Top = 42
Width = 24
Height = 13
Caption = 'Units'
end
<> object Label4: TLabel
Left = 198
Top = 78
Width = 64
Height = 13
Caption = 'LargeChange'
end
<> object Label5: TLabel
Left = 242
Top = 117
Width = 20
Height = 13
Caption = 'Max'
end
<> object Label1: TLabel
Left = 280

TAboutBox = class(TForm)
Button1: TButton;
Label1: TLabel;
Label2: TLabel;
<> procedure
Button1Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
AboutBox: TAboutBox;
implementation
{\$R *.DFM}
<> procedure
TAboutBox.Button1Click(Sender:
TObject);
begin
Close;
end;
end.

الملف النصى للبرنامج
Scroll bar
وفيه
Objects 40
<u>من إعداد</u>
علاء الدين اللباد
. 9 £ £ 0 V 0 T V 1
<> object ScrollForm1:
TScrollForm1
Left = 181
Top = 156
BorderIcons = [biSystemMenu,

ReadOnly = True
TabOrder = 4
Text = '0'
end
<> object HMax: TEdit
Left = 276
Top = 113 Width = 48
Height = 21
ReadOnly = True
TabOrder = 5
Text = '0'
end
<> object VUnits: TEdit
Left = 363
Top = 38
Width = 48
Height = 21
ReadOnly = True
TabOrder = 6
Text = '1'
end
<> object VPage: TEdit
Left = 363
Top = 74
Width = 48
Height = 21
ReadOnly = True
TabOrder = 7
Text = '0'
end
<> object VMax: TEdit
Left = 363
Top = 113
Width = 48
Height = 21
ReadOnly = True
TabOrder = 8
Text = '0'
end
<> object HUnitsUpDown:
TUpDown
Left = 323
Top = 38 Width = 15
Height = 21
Associate = HUnits
Min = 1

Top = 16
Width = 47
Height = 13
Caption = 'Horizontal'
end
<> object Vertical: TLabel
Left = 376
Top = 16
Width = 35
Height = 13
Caption = 'Vertical'
end
<> object HScrollb:
TScrollBar
Left = 9
Top = 169
Width = 159
Height = 16
Max = 32767
PageSize = 0
TabOrder = 1
OnScroll = HScrollbScroll
end
<> object VScrollb:
TScrollBar
Left = 169
Top = 9
Width = 16
Height = 157
Kind = sbVertical
PageSize = 0
TabOrder = 2
OnChange = VScrollbChange
end
<> object HUnits: TEdit
Left = 275
Top = 38
Width = 48
Height = 21
ReadOnly = True
TabOrder = 3
Text = '1'
end
<> object HPage: TEdit
Left = 275
Top = 74
Width = 48
Height = 21
1101giit - 21

TabOrder = 13
OnClick = UpDownClick
end
<> object VPageUpDown:
TUpDown
Left = 411
Top = 74
Width = 15
Height = 21
Associate = VPage
Max = 32767
TabOrder = 14
OnClick = UpDownClick
end
<> object DefaultBtn:
TButton
Left = 348
Top = 157
Width = 75
Height = 25
Caption = '&Defaults'
ParentShowHint = False
ShowHint = False
TabOrder = 15
OnClick = DefaultBtnClick
end
<> object ApplyBtn: TButton
Left = 265
Top = 157
Width = 75
Height = 25
Caption = '&Apply'
Enabled = False
TabOrder = 16
OnClick = ApplyBtnClick
end
<> object ImgPanel: TPanel
Left = 8
Top = 8
Width = 160
Height = 160
BevelOuter = bvNone
BorderStyle = bsSingle
TabOrder = 0
<> object Image: TImage
Left = 0
Top = 0
Width = 600

Max = 32767
Position = 1
TabOrder = 9
OnClick = UpDownClick
end
<> object HPageUpDown:
TUpDown
Left = 323
Top = 74 Width = 15
Height = 21
Associate = HPage
$\mathbf{Max} = 32767$
TabOrder = 10
OnClick = UpDownClick
end
<> object HMaxUpDown:
TUpDown
Left = 324
Top = 113
Width = 15
Height = 21
Associate = HMax
Max = 32767
TabOrder = 11
OnClick = UpDownClick
end
<> object VUnitsUpDown:
TUpDown
Left = 411
Top = 38
Width = 15
Height = 21
Associate = VUnits
Min = 1
Max = 32767
Position = 1
TabOrder = 12
OnClick = UpDownClick
end
<> object VMaxUpDown:
TUpDown
Left = 411
Top = 113
Width = 15
Height = 21
Associate = VMax
Max = 32767

end
end
end
<> object OpenDialog:
TOpenDialog
DefaultExt = '.bmp' Filter =
'Images
(*.bmp;*.ico) *.bmp;*.ico Bitmaps
(*.bmp) *.bmp Icons (*.' +
'ico) *.ico'
Options = [ofHideReadOnly,
ofPathMustExist, ofFileMustExist]
Left = 230
Top = 158
end
End
<> object AboutBox:
TAboutBox
Left = 200
Top = 108
BorderIcons = [biSystemMenu]
BorderStyle = bsDialog
Caption = 'About ScrollDemo'
ClientHeight = 135
ClientWidth = 239
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
PixelsPerInch = 96
TextHeight = 13
<> object Label1: TLabel
Left = 24
Top = 24
Width = 191
Height = 20
Caption = 'ScrollBar
Demonstation'
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
I UIII CUIUI CI II IIIUU II I CAL

Height = 470
AutoSize = True
Picture.Data = {
}
end
end
<> object MainMenu1:
TMainMenu
Left = 196
Top = 159
<> object File1: TMenuItem
Caption = '&File'
<> object Open1:
TMenuItem
Caption = '&Open'
OnClick = Open1Click
end
<> object N1: TMenuItem
Caption = '-'
end
<> object Exit1:
TMenuItem
Caption = 'E&xit'
OnClick = Exit1Click
end
end
<> object View1:
TMenuItem
Caption = '&Scroll'
OnClick = View1Click
<> object VScrollMenu:
TMenuItem
Caption = '&Vertical'
OnClick = VScrollMenuClick
end
<> object HScrollMenu:
TMenuItem
Caption = '&Horizontal'
OnClick = HScrollMenuClick
end
end
<> object Help1:
TMenuItem
Caption = '&Help'
<> object About1:
TMenuItem
Caption = '&About'
•
OnClick = About1Click

TMainForm = class(TForm) ApplicationEvents: TApplicationEvents; ActionList: TActionList; Action: TAction; MainMenu: TMainMenu; MenuAction: TMenuItem; MenuException: TMenuItem; HintButton: TButton; lbOnMessage: TListBox; lblOnMessage: TLabel; lbOther: TListBox; lblOther: TLabel; lbIdle: TListBox; lblOnIdle: TLabel; lbActionUpdate: TListBox; lblOnActionUpdate: TLabel; procedure ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean); procedure
TMainForm = class(TForm) ApplicationEvents: TApplicationEvents; ActionList: TActionList; Action: TAction; MainMenu: TMainMenu; MenuAction: TMenuItem; MenuException: TMenuItem; HintButton: TButton; IbOnMessage: TListBox; IbIOnMessage: TLabel; IbOther: TListBox; IbIOther: TLabel; IbIdle: TListBox; IbIOnIdle: TLabel; IbActionUpdate: TListBox; IbIOnActionUpdate: TLabel; procedure ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean);
ApplicationEvents: TApplicationEvents; ActionList: TActionList; Action: TAction; MainMenu: TMainMenu; MenuAction: TMenuItem; MenuException: TMenuItem; HintButton: TButton; IbOnMessage: TListBox; IblOnMessage: TLabel; IbOther: TListBox; IblOther: TLabel; IbIdle: TListBox; IblOnIdle: TLabel; IbActionUpdate: TListBox; IblOnActionUpdate: TLabel; procedure ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean);
TApplicationEvents; ActionList: TActionList; Action: TAction; MainMenu: TMainMenu; MenuAction: TMenuItem; MenuException: TMenuItem; HintButton: TButton; IbOnMessage: TListBox; IblOnMessage: TLabel; IbOther: TListBox; IblOther: TLabel; IbIdle: TListBox; IblOnIdle: TLabel; IbActionUpdate: TListBox; IblOnActionUpdate: TLabel; procedure ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean);
ActionList: TActionList; Action: TAction; MainMenu: TMainMenu; MenuAction: TMenuItem; MenuException: TMenuItem; HintButton: TButton; IbOnMessage: TListBox; IbIOnMessage: TLabel; IbOther: TListBox; IbIOther: TLabel; IbIdle: TListBox; IbIOnIdle: TLabel; IbActionUpdate: TListBox; IbIOnActionUpdate: TLabel; procedure ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean);
Action: TAction; MainMenu: TMainMenu; MenuAction: TMenuItem; MenuException: TMenuItem; HintButton: TButton; IbOnMessage: TListBox; IbIOnMessage: TLabel; IbOther: TListBox; IbIOther: TLabel; IbIdle: TListBox; IbIOnIdle: TLabel; IbActionUpdate: TListBox; IbIOnActionUpdate: TLabel; procedure ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean);
MainMenu: TMainMenu; MenuAction: TMenuItem; MenuException: TMenuItem; HintButton: TButton; IbOnMessage: TListBox; IbIOnMessage: TLabel; IbOther: TListBox; IbIOther: TLabel; IbIdle: TListBox; IbIOnIdle: TLabel; IbActionUpdate: TListBox; IbIOnActionUpdate: TLabel; procedure ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean);
MenuAction: TMenuItem; MenuException: TMenuItem; HintButton: TButton; IbOnMessage: TListBox; IbIOnMessage: TLabel; IbOther: TListBox; IbIOther: TLabel; IbIdle: TListBox; IbIOnIdle: TLabel; IbActionUpdate: TListBox; IbIOnActionUpdate: TLabel; procedure ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean);
MenuException: TMenuItem; HintButton: TButton; IbOnMessage: TListBox; IbIOnMessage: TLabel; IbOther: TListBox; IbIOther: TLabel; IbIdle: TListBox; IbIOnIdle: TLabel; IbActionUpdate: TListBox; IbIOnActionUpdate: TLabel; procedure ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean);
HintButton: TButton; IbOnMessage: TListBox; IbIOnMessage: TLabel; IbOther: TListBox; IbIOther: TLabel; IbIdle: TListBox; IbIOnIdle: TLabel; IbActionUpdate: TListBox; IbIOnActionUpdate: TLabel; procedure ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean);
lbOnMessage: TListBox; lblOnMessage: TLabel; lbOther: TListBox; lblOther: TLabel; lbIdle: TListBox; lblOnIdle: TLabel; lbActionUpdate: TListBox; lblOnActionUpdate: TLabel; procedure ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean);
IblOnMessage: TLabel; IbOther: TListBox; IblOther: TLabel; IbIdle: TListBox; IblOnIdle: TLabel; IbActionUpdate: TListBox; IblOnActionUpdate: TLabel; procedure ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean);
lbOther: TListBox; lbIOther: TLabel; lbIdle: TListBox; lbIOnIdle: TLabel; lbActionUpdate: TListBox; lbIOnActionUpdate: TLabel; procedure ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean);
IbIOther: TLabel; IbIdle: TListBox; IbIOnIdle: TLabel; IbActionUpdate: TListBox; IbIOnActionUpdate: TLabel; procedure ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean);
IbIdle: TListBox; IblOnIdle: TLabel; IbActionUpdate: TListBox; IblOnActionUpdate: TLabel; procedure ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean);
lblOnIdle: TLabel; lbActionUpdate: TListBox; lblOnActionUpdate: TLabel; procedure ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean);
lbActionUpdate: TListBox; lblOnActionUpdate: TLabel; procedure ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean);
IblOnActionUpdate: TLabel; procedure ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean);
procedure ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean);
ApplicationEventsActionExecute(Action: TBasicAction; var Handled: Boolean);
tion: TBasicAction; var Handled: Boolean);
var Handled: Boolean);
,
procedure
-
ApplicationEventsActionUpdate(Act
ion: TBasicAction;
var Handled: Boolean);
procedure
ApplicationEventsActivate(Sender:
TObject);
procedure
ApplicationEventsDeactivate(Sender
: TObject);
procedure
ApplicationEventsException(Sender
: TObject; E: Exception);
function
ApplicationEventsHelp(Command:
Word; Data: Integer;
var CallHelp: Boolean): Boolean;
procedure
ApplicationEventsHint(Sender:
TObject);
procedure
ApplicationEventsIdle(Sender:
TObject; var Done: Boolean);
procedure
ApplicationEventsMessage(var Msg:
tagMSG;

Font.Height = -16
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
<> object Label2: TLabel
Left = 32
Top = 56
Width = 174
Height = 13
Caption = 'Copyright 1997,
Borland International'
end
<> object Button1: TButton
Left = 82
Top = 96
Width = 75
Height = 25
Caption = 'OK'
TabOrder = 0
OnClick = Button1Click
end
end

# Objects الملف لبرنامج (AppEvents) المعاد إعداد إعداد بعداد بعدا

OnActionUpdate');	var Handled: Boolean);		
end;	procedure		
CHU,	ApplicationEventsMinimize(Sender:		
procedure	TObject);		
-	procedure		
TMainForm.ApplicationEventsActiv	<del>-</del>		
ate(Sender: TObject);	ApplicationEventsRestore(Sender:		
begin	TObject);		
lbOther.Items.Add('Event	procedure		
OnActivate');	ApplicationEventsShortCut(var		
end;	Msg: TWMKey;		
	var Handled: Boolean);		
procedure	procedure		
TMainForm.ApplicationEventsDeac	ApplicationEventsShowHint(var		
tivate(Sender: TObject);	HintStr: String;		
begin	var CanShow: Boolean; var		
lbOther.Items.Add('Event	HintInfo: THintInfo);		
OnDeactivate');	procedure ActionExecute(Sender:		
end;	TObject);		
	procedure		
procedure	MenuExceptionClick(Sender:		
TMainForm.ApplicationEventsExce	TObject);		
	private		
ption(Sender: TObject;	{ Private declarations }		
E: Exception);			
begin	public (Parking de dans de la continua )		
lbOther.Items.Add('Event	{ Public declarations }		
OnException: ' + E.Message);	end;		
end;			
	var		
function	MainForm: TMainForm;		
TMainForm.ApplicationEventsHelp			
(Command: Word; Data: Integer;	implementation		
var CallHelp: Boolean): Boolean;			
begin	{\$R *.dfm}		
lbOther.Items.Add('Event			
OnHelp');	procedure		
Result := False;	TMainForm.ApplicationEventsActio		
end;	nExecute(Action: TBasicAction;		
	var Handled: Boolean);		
procedure	begin		
TMainForm.ApplicationEventsHint(	lbOther.Items.Add('Event		
= = :	OnActionExecute');		
Sender: TObject);	, ,		
begin	end;		
lbOther.Items.Add('Event			
OnHint');	procedure		
end;	TMainForm.ApplicationEventsActio		
	nUpdate(Action: TBasicAction;		
procedure	var Handled: Boolean);		
TMainForm.ApplicationEventsIdle(	begin		
Sender: TObject;	lbActionUpdate.Items.Add('Event		

procedi	ure
-	Form.ActionExecute(Sender:
TObjec	
begin	
Show	Message('Action executed');
end;	
procedi	ure
TMain]	Form.MenuExceptionClick(S
ender:	TObject);
resourc	estring
sExce	otionRaised = 'This is an
exception	on';
begin	
raise	
Excepti	on.Create(sExceptionRaised)
;	
end;	
end.	

الملف النصى للبرنامج
Scroll bar
وفيه
Objects 40
من إعداد
علاء الدين اللباد
.9 2 2 0 0 0 7 0 1
<> object ScrollForm1:
TScrollForm1
Left = 181
Top = 156
BorderIcons = [biSystemMenu,
biMinimize]
BorderStyle = bsSingle
Caption = 'ScrollDemo'
ClientHeight = 194
ClientWidth = 436
Color = clBtnFace

var Done: Boolean);
begin
lbIdle.Items.Add('Event OnIdle');
end;
procedure
TMainForm.ApplicationEventsMess
age(var Msg: tagMSG;
var Handled: Boolean);
begin
lbOnMessage.Items.Add('X:=' +
IntToStr(Msg.pt.x) + ' Y:=' +
IntToStr(Msg.pt.y));
end;
_
procedure
TMainForm.ApplicationEventsMini
mize(Sender: TObject);
begin
lbOther.Items.Add('Event
OnMinimize');
end;
procedure
TMainForm.ApplicationEventsRest
ore(Sender: TObject);
begin
lbOther.Items.Add('Event
OnRestore');
end;
procedure TMoinForm ApplicationEventsShore
TMainForm.ApplicationEventsShor
tCut(var Msg: TWMKey; var Handled: Boolean);
begin
lbOther.Items.Add('Event
OnShortCut');
end;
ciu,
procedure
TMainForm.ApplicationEventsShow
Hint(var HintStr: String;
var CanShow: Boolean; var
HintInfo: THintInfo);
begin
lbOther.Items.Add('Event
OnShowHint');
end;

T 6: 250
Left = 376
Top = 16
Width = 35
Height = 13
Caption = 'Vertical'
end
<> object HScrollb:
TScrollBar
Left = 9
Top = 169
Width = 159
Height = 16
Max = 32767
PageSize = 0
TabOrder = 1
OnScroll = HScrollbScroll
end
<> object VScrollb:
TScrollBar
Left = 169
Top = 9
Width = 16
Height = 157
Kind = sbVertical
PageSize = 0
TabOrder = 2
OnChange = VScrollbChange
end
<> object HUnits: TEdit
Left = 275
Top = 38
Width = 48
Height = 21
ReadOnly = True
TabOrder = 3
Text = '1'
end
<> object HPage: TEdit
Left = 275
Top = 74
Width = 48
Height = 21
ReadOnly = True
TabOrder = 4
Text = '0'
end
<> object HMax: TEdit
Left = 276

Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
Menu = MainMenu1
OldCreateOrder = True
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Bevel1: TBevel
Left = 0
Top = 0
Width = 436
Height = 9
Align = alTop
Shape = bsTopLine
end
<> object Label3: TLabel
Left = 238
Top = 42
Width = 24
Height = 13
Caption = 'Units'
end
<> object Label4: TLabel
Left = 198
Top = 78
Width = 64
Height = 13
Caption = 'LargeChange'
end
<> object Label5: TLabel
Left = 242
Top = 117
Width = 20
Height = 13
Caption = 'Max'
end
<> object Label1: TLabel
Left = 280
Top = 16 Width = 47
Height = 13
Caption = 'Horizontal'
end
<> object Vertical: TLabel

TUpDown
Left = 323
Top = 74
Width = 15
Height = 21
Associate = HPage
Max = 32767
TabOrder = 10
OnClick = UpDownClick
end
<> object HMaxUpDown:
TUpDown
Left = 324
Top = 113
Width = 15
Height = 21
Associate = HMax
Max = 32767
Max - 52/0/ TabOrder = 11
OnClick = UpDownClick end
<> object VUnitsUpDown:
TUpDown Left = 411
Top = 38 Width = 15
Height = 21
Associate = VUnits
Min = 1
Max = 32767
Position = 1
TabOrder = 12
OnClick = UpDownClick
end
<> object VMaxUpDown:
TUpDown
Left = 411
Top = 113
Width = 15
Height = 21
Associate = VMax
Max = 32767
TabOrder = 13
OnClick = UpDownClick
end
<> object VPageUpDown:
TUpDown
Left = 411

Top = 113
Width = 48
Height = 21
ReadOnly = True
TabOrder = 5
Text = '0'
end
<> object VUnits: TEdit Left = 363
Top = 38
Width = 48
Height = 21
ReadOnly = True
TabOrder = 6
Text = '1'
end
<> object VPage: TEdit
Left = 363
Top = 74
Width = 48
Height = 21
ReadOnly = True
TabOrder = 7
Taborder - 7 Text = '0'
end
<> object VMax: TEdit
Left = 363
Top = 113
Width = 48
Height = 21
ReadOnly = True
TabOrder = 8
Text = '0'
end
<> object HUnitsUpDown:
TUpDown
Left = 323
Top = 38
Width = 15
Height = 21
Associate = HUnits
Min = 1
Max = 32767
Position = 1
TabOrder = 9
OnClick = UpDownClick
end
<> object HPageUpDown:
g 9 11 11 1

<> object MainMenu1:
TMainMenu
Left = 196
Top = 159
<> object File1: TMenuItem
Caption = '&File'
<> object Open1:
TMenuItem
Caption = '&Open'
OnClick = Open1Click
end
<> object N1: TMenuItem
Caption = '-'
end
<> object Exit1:
TMenuItem
Caption = 'E&xit'
OnClick = Exit1Click
end
end
<> object View1:
TMenuItem
Caption = '&Scroll'
OnClick = View1Click
<> object VScrollMenu:
TMenuItem
Caption = '&Vertical'
OnClick = VScrollMenuClick
end
<> object HScrollMenu:
TMenuItem
Caption = '&Horizontal'
OnClick = HScrollMenuClick
end
end
<> object Help1:
TMenuItem
Caption = '&Help'
<> object About1:
TMenuItem
Caption = '&About'
OnClick = About1Click
end
end
end
<> object OpenDialog:
TOpenDialog
DefaultExt = '.bmp'

Top = 74
Width = 15
Height = 21
Associate = VPage
Max = 32767
TabOrder = 14
OnClick = UpDownClick
end
<> object DefaultBtn:
TButton
Left = 348
Top = 157
Width = 75
Height = 25
Caption = '&Defaults'
ParentShowHint = False
ShowHint = False
TabOrder = 15
OnClick = DefaultBtnClick
end
<> object ApplyBtn: TButton
Left = 265
Top = 157
Width = 75
Height = 25
Caption = '&Apply'
Enabled = False
TabOrder = 16
OnClick = ApplyBtnClick
end
<> object ImgPanel: TPanel
Left = 8
Top = 8
Width = 160
Height = 160
BevelOuter = bvNone
BorderStyle = bsSingle
TabOrder = 0
<> object Image: TImage
Left = 0
Top = 0
-
Width = 600
Height = 470
AutoSize = True
Picture.Data = {
}
end
end

Left = 32
Top = 56
Width = 174
Height = 13
Caption = 'Copyright 1997,
<b>Borland International'</b>
end
<> object Button1: TButton
Left = 82
Top = 96
Width = 75
Height = 25
Caption = 'OK'
TabOrder = 0
OnClick = Button1Click
end
end

Filter =
'Images
(*.bmp;*.ico) *.bmp;*.ico Bitmaps
(*.bmp) *.bmp Icons (*.' +
'ico) *.ico'
Options = [ofHideReadOnly,
ofPathMustExist, ofFileMustExist
Left = 230
Top = 158
end
End
Enu
i i i i i i
<> object AboutBox:
TAboutBox
Left = 200
Top = 108
BorderIcons = [biSystemMenu]
BorderStyle = bsDialog
Caption = 'About ScrollDemo'
ClientHeight = 135
ClientWidth = 239
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
PixelsPerInch = 96
TextHeight = 13
<> object Label1: TLabel
Left = 24
Top = 24
Width = 191
Height = 20
Caption = 'ScrollBar
Demonstation'
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font Name = IMS Sans Sariff
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
<> object Label2: TLabel

	Object8
{\$R *.dfm}	الملف التنفيذي لبرنامج
uses	(ConvertI)
ConvUtils, StdConvs, EuroConv,	
StrUtils;	إعداد
procedure	علاء الدين اللباد
TForm1.FormShow(Sender:	
TObject);	هانف ۱۷۳۵۷۵۶۹۰
var	unit ConvertItUnit;
LFamilies: TConvFamilyArray;	and converse may
I: Integer;	interface
LStrings: TStringList;	
begin	uses
ConvFamilies.Tabs.Clear;	Windows, Messages, SysUtils,
LStrings := TStringList.Create;	Variants, Classes, Graphics,
try	Controls, Forms,
GetConvFamilies(LFamilies);	Dialogs, StdCtrls, ComCtrls;
for I := 0 to Length(LFamilies) - 1	
do	type
	TForm1 = class(TForm)
LStrings.AddObject(ConvFamilyTo	ConvTypes: TListBox;
Description(LFamilies[I]),	ConvValue: TEdit;
TObject(LFamilies[I]));	ConvResults: TListBox;
LStrings.Sort;	ConvValueIncDec: TUpDown;
	ConvFamilies: TTabControl;
ConvFamilies.Tabs.Assign(LStrings)	StatusBar1: TStatusBar;
;	procedure FormShow(Sender:
ConvFamiliesChange(Sender);	TObject);
finally	procedure
LStrings.Free;	ConvTypesClick(Sender: TObject);
end;	procedure
end;	ConvValueChange(Sender:
	TObject);
procedure	procedure
TForm1.ConvFamiliesChange(Send	ConvFamiliesChange(Sender:
er: TObject);	TObject);
var	private
LFamily: TConvFamily;	{ Private declarations }
LTypes: TConvTypeArray;	public
I: Integer;	{ Public declarations }
begin	end;
LFamily :=	
TConvFamily(ConvFamilies.Tabs.O	var
bjects[ConvFamilies.TabIndex]);	Form1: TForm1;
with ConvTypes, Items do	
begin	implementation
BeginUpdate;	

LTestType),
ConvTypeToDescription(LTestType
)[));
end;
end
else
Add('No base type');
except
Add('Cannot parse value');
end;
finally
EndUpdate;
end;
end;
end.
_

Objec	ts
	الهاف النصي لبرنامج
	(ConvertI)
	alaeļ
	علاء الدين اللباد
	واتف ۷۹۲۵۷۵۲۹۰
object l	Form1: TForm1
Left =	192
Top =	107
Width	$_{1} = 420$
	t = 466
Captio	on = 'Conversion Tester'
	= clBtnFace
Font.	Charset =
DEFAU	JLT_CHARSET
Font.	Color = clWindowText
Font.I	Height = -11
Font.N	Name = 'MS Sans Serif'
Font.S	Style = []
OldCı	reateOrder = False
OnSh	ow = FormShow
Pixels	PerInch = 96
TextH	leight = 13
object	<b>ConvFamilies: TTabControl</b>

Clear;
GetConvTypes(LFamily, LTypes);
for I := 0 to Length(LTypes) - 1 do
AddObject(ConvTypeToDescription
(LTypes[I]), TObject(LTypes[I]));
ItemIndex := 0;
EndUpdate;
end:
ConvTypesClick(Sender);
end;
procedure
TForm1.ConvTypesClick(Sender:
TObject);
begin
ConvValueChange(Sender);
end;
enu,
nunanduun
procedure
TForm1.ConvValueChange(Sender:
TObject);
var LV L D LL
LValue: Double;
LBaseType, LTestType:
TConvType;
I: Integer;
begin
with ConvResults, Items do
try
BeginUpdate;
Clear;
try
LValue :=
StrToFloatDef(ConvValue.Text, 0);
if ConvTypes.ItemIndex <> -1
then
begin
LBaseType :=
TConvType(ConvTypes.Items.Objec
ts[ConvTypes.ItemIndex]);
for $I := 0$ to
ConvTypes.Items.Count - 1 do
begin
LTestType :=
TConvType(ConvTypes.Items.Objec
ts[I]);
Add(Format('%n %s',
[Convert(LValue, LBaseType,
1 converge variety District ype,

ItemHeight = 13
ParentColor = True
TabOrder = 3
end
object ConvValue: TEdit
Left = 200
Top = 32
Width = 177
Height = 21
Anchors = [akLeft, akTop,
akRight]
TabOrder = 1
Text = '1'
OnChange = ConvValueChange
end
end
object StatusBar1: TStatusBar
Left = 0
Top = 413
Width = 412
Height = 19
Panels = <
item
Width = 50
end>
end
End

Left = 0
Top = 0
Width = 412
Height = 413
Align = alClient
TabOrder = 0
Tabs.Strings = (
'Testing')
TabIndex = 0
OnChange =
ConvFamiliesChange
DesignSize = (
412
413)
object ConvTypes: TListBox
Left = 16
Top = 32
Width = 177
Height = 372
3
Anchors = [akLeft, akTop,
akBottom]
ItemHeight = 13
Sorted = True
TabOrder = 0
OnClick = ConvTypesClick
end
object ConvValueIncDec:
TUpDown
Left = 377
Top = 32
Width = 16
Height = 21
Anchors = [akTop, akRight]
Associate = ConvValue
Min = -32000
Max = 32000
Position = 1
TabOrder = 2
Thousands = False
end
object ConvResults: TListBox
Left = 200
Top = 56
Width = 196
Height = 348
TabStop = False
Anchors = [akLeft, akTop,
akRight, akBottom]

procedure	
TrackBar1Change(Sender:	الملف التنفيذي لبرنامج
TObject);	
procedure TrackBar2Change(Sender:	(Prgrsba)
TObject);	عامدا
procedure FormDestroy(Sender:	AIAE
TObject);	علاء الدين اللباد
private	
{ Private declarations }	هاتف ۷۱۳۵۷۵۶۶۰
MyThread1 : TMyThread; //	unit Pg1;
thread number 1	unit i gi,
MyThread2 : TMyThread; //	Interface
thread number 2	Interface
Thread1Active: boolean; // used	Uses
to test if thread 1 is active	Windows, Messages, SysUtils,
Thread2Active: boolean; // used	Classes, Graphics, Controls, Forms,
to test if thread 2 is active	Dialogs,
procedure ThreadDone(var	StdCtrls, ComCtrls, ExtCtrls, Pg2;
AMessage : TMessage); message	
WM_ThreadDoneMsg; // Message to	Const
be sent back from thread when its done	WM_ThreadDoneMsg = WM_User
public	+8;
{ Public declarations }	
end;	Туре
chu,	TForm1 = class(TForm)
var	ProgressBar1: TProgressBar;
Form1: TForm1;	ProgressBar2: TProgressBar;
	Button1: TButton;
implementation	Button2: TButton; TrackBar1: TTrackBar;
•	TrackBar1: 11rackBar; TrackBar2: TTrackBar;
{\$R *.DFM}	Bevel1: TBevel;
	Bevel2: TBevel;
procedure	Label1: TLabel;
TForm1.Button1Click(Sender:	Label: TLabel;
TObject); // Create Thread 1	Button3: TButton;
{ The thread will destroy iteself	Button4: TButton;
when it is done executing because	procedure Button1Click(Sender:
FreeOnTerminate is set to true.	TObject);
The first paramter is the priority,	procedure Button2Click(Sender:
and the second is the progressbar to	TObject);
update.	procedure Button3Click(Sender:
begin	TObject);
if (MyThread1 = nil) or	procedure Button4Click(Sender:
(Thread1Active = false) then // make	TObject);
sure its not already running	procedure FormCreate(Sender:
begin	TObject);
~ -8	

begin	MyThread2.Terminate
TObject); // set Thread 2 Priority	to see if it is running
TForm1.TrackBar2Change(Sender:	(Thread2Active = true) then // check
procedure	if (MyThread2 <> nil) and
,	Begin
end;	TObject); // Terminate Thread 2
n);	TForm1.Button4Click(Sender:
TThreadPriority(TrackBar1.Positio	procedure TEarm 1 Putton 4 Click (Sandon)
MyThread1.priority :=	nuacaduna
(Thread1Active = true) then	end;
if (MyThread1 $\Leftrightarrow$ nil) and	started');
begin	S (
TObject); // set Thread 1 Priority	ShowMessage('Thread not
TForm1.TrackBar1Change(Sender:	Else
procedure	MyThread1.Terminate
nraadura	to see if it is running
	(Thread1Active = true) then // check
Ciru,	if (MyThread1 <> nil) and
end;	Begin
Thread2Active := false;	TObject); // Terminate Thread 1
Thread1Active := false;	TForm1.Button3Click(Sender:
begin	procedure
TObject); // initialize to zero	
TForm1.FormCreate(Sender:	end;
procedure	executing');
	ShowMessage('Thread still
	Else
end;	End
end;	Thread2Active := true;
Thread2Active := false;	sition, ProgressBar2);
begin	TMyThread.CreateIt(TrackBar2.Po
cardinal(AMessage.WParam))) then	MyThread2 :=
(MyThread2.ThreadID =	Begin
if ((MyThread2 <> nil) and	make sure its not already running
end;	(Thread2Active = false) then //
Thread1Active := false;	if (MyThread2 = nil) or
begin	Begin
cardinal(AMessage.WParam))) then	TObject); // Create Thread 2
(MyThread1.ThreadID =	TForm1.Button2Click(Sender:
if ((MyThread1 <> nil) and	procedure
begin	
executing	end;
of when and which thread is done	executing');
AMessage: TMessage); // keep track	ShowMessage('Thread still
procedure TForm1.ThreadDone(var	Else
	End
end;	Thread1Active := true;
started');	sition, ProgressBar1);
ShowMessage('Thread not	TMyThread.CreateIt(TrackBar1.Po
Else	MyThread1 :=
	7.5 (77)

Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = False
Position = poDesktopCenter
OnCreate = FormCreate
OnDestroy = FormDestroy
PixelsPerInch = 96
TextHeight = 13
object Bevel1: TBevel
Left = 32
Top = 8
Width = 513
Height = 169
end
object Bevel2: TBevel
Left = 32
Top = 200
Width = 513
Height = 169
end
object Label1: TLabel
Left = 256
Top = 104
Width = 34
Height = 13
Caption = 'Priority:'
end
object Label2: TLabel
Left = 248
Top = 296
Width = 34
Height = 13
Caption = 'Priority:'
end
object ProgressBar1: TProgressBar
Left = 64
Top = 32
Width = 393
Height = 25
Step = 1
TabOrder = 0
end
object ProgressBar2: TProgressBar

if (MyThread2 <> nil) and
(Thread2Active = true) then
MyThread2.priority :=
TThreadPriority(TrackBar2.Positio
n);
end;
procedure
TForm1.FormDestroy(Sender:
TObject); // Terminate any threads
still running
Begin
if (MyThread1 <> nil) and
(Thread1Active = true) then
Begin
MyThread1.Terminate;
MyThread1.WaitFor; // wait for
it to terminate
end;
if (MyThread2 <> nil) and
(Thread2Active = true) then
Begin
MyThread2.Terminate;
MyThread2.WaitFor;
end;
end;
end.

<b>ObjectS</b>
الهلف النصي لبرناهج
(Prgrsba)
إعداد
علاء الدين اللباد
واتف ۷۱۳۵۷۵۶۹۰
object Form1: TForm1
Left = 169
Top = 109
Width = 712
Height = 507
Caption = 'TThread, TTrackBar,
TProgressBar'

Height = 25
Caption = 'Terminate Thread'
TabOrder = 6
OnClick = Button3Click
end
object Button4: TButton
Left = 72
Top = 320
Width = 105
Height = 25
Caption = 'Terminate Thread'
TabOrder = 7
OnClick = Button4Click
end
End

Left = 64
Top = 224
Width = 401
Height = 25
TabOrder = 1
end
object Button1: TButton
Left = 72
Top = 80
Width = 105
Height = 25
Caption = 'Create Thread'
TabOrder = 2
OnClick = Button1Click
end
object Button2: TButton Left = 72
Top = 272 Width = 105
Height = 25
Caption = 'Create Thread'
TabOrder = 3
OnClick = Button2Click
end
object TrackBar1: TTrackBar
Left = 296
Top = 104
Width = 193
Height = 33
Max = 6
Position = 3
TabOrder = 4
OnChange = TrackBar1Change
end
object TrackBar2: TTrackBar
Left = 288
Top = 296
Width = 193
Height = 45
Max = 6
Position = 3
TabOrder = 5
OnChange = TrackBar2Change
end
object Button3: TButton
Left = 72
Top = 128
Width = 105

<pre>procedure Button1Click(Sender: TObject);</pre>
procedure Button2Click(Sender:
TObject);
procedure Button3Click(Sender:
TObject);
procedure
LCDNumber1MouseUp(Sender:
TObject; Button: TMouseButton;
Shift: TShiftState; X, Y:
Integer);
private
{ Private declarations }
FRunning: Boolean;
procedure SetRunning(Value:
Boolean);
public
{ Public declarations }
StartTime: Extended;
ElapsedTime: Extended;
Reset: Boolean;
property Running: Boolean read
FRunning write SetRunning;
end;
var
Form1: TForm1;
implementation
-
{\$R *.xfm}
procedure
TForm1.SetRunning(Value:
Boolean);
begin
if FRunning <> Value then
begin
FRunning := Value;
StartTime := ElapsedTime;
if Value then Reset := False;
end;
end;
,
procedure
TForm1.Timer1Timer(Sender:
TObject);
begin
if Reset then ElapsedTime := Now
11011

الملف التنفيذي لبرنامج	
(StopWatch)	5
عامدا	
علاء الدين اللباد	
41 الم	
program stopwatch;	
uses	
QForms,	
main in 'main.pas' {Form1};	
{\$R *.res}	_
begin Application.Initialize;	
Application.CreateForm(TForm1,	
Form1);	
Application.Run; end.	
enu.	
unit main;	
interface	_
SysUtils, Types, Classes,	
QGraphics, QControls, QForms,	
QDialogs,	
QStdCtrls, QTypes, QExtCtrls;	
type	_
TForm1 = class(TForm)	
LCDNumber1: TLCDNumber;	
Timer1: TTimer; Button1: TButton;	_
Button2: TButton;	_
Button3: TButton;	_
<pre>procedure Timer1Timer(Sender: TObject);</pre>	
1 Object),	_

if NewBorderStyle >			
Ord(High(TBorderStyle)) then			
NewBorderStyle :=			
Ord(Low(TBorderStyle));			
LCDNumber1.BorderStyle :=			
TBorderStyle(NewBorderStyle);			
end;			
mbRight:			
begin			
NewSegmentStyle :=			
Ord(LCDNumber1.SegmentStyle) +			
1;			
if NewSegmentStyle >			
Ord(High(TLCDSegmentStyle))			
then			
NewSegmentStyle :=			
Ord(Low(TLCDSegmentStyle));			
LCDNumber1.SegmentStyle :=			
TLCDSegmentStyle(NewSegmentSt			
yle);			
end;			
end;			
end;			
end.			

<b>ObjectS</b>	
الملف لبرنامج	
(StopWatch)	
إعداد	
علاء الدين اللباد	
واتف ۷۹۲۵۷۵۴۹۰	
object Form1: TForm1	
Left = 117	
Top = 167	
<b>Width = 237</b>	
Height = 67	
VertScrollBar.Range = 65	
HorzScrollBar.Range = 233	
ActiveControl = Button1	
Caption = 'Form1'	
Color = clBackground	
PixelsPerInch = 75	
TextHeight = 13	

also FlorgodTime - Novy				
else ElapsedTime := Now -				
StartTime;				
if Running then				
LCDNumber1.Value :=				
FormatDateTime('nn:ss.zzz',				
ElapsedTime);				
end;				
procedure				
1 -				
TForm1.Button1Click(Sender:				
TObject);				
begin				
Running := True;				
end;				
procedure				
TForm1.Button2Click(Sender:				
TObject);				
begin				
8				
Running := False;				
end;				
procedure				
TForm1.Button3Click(Sender:				
TObject);				
begin				
if not Running then				
begin				
LCDNumber1.Value :=				
'00:00.000';				
· · · · · · · · · · · · · · · · · · ·				
Reset := True;				
end;				
end;				
procedure				
TForm1.LCDNumber1MouseUp(Se				
nder: TObject; Button:				
TMouseButton;				
Shift: TShiftState; X, Y: Integer);				
var				
NewBorderStyle: Integer;				
NewSegmentStyle: Integer;				
begin				
case Button of				
mbLeft:				
begin				
NewBorderStyle :=				
Ord(LCDNumber1.BorderStyle) +				
1;				
<b>1</b> 9				

end	
End	

TextWidth = 6
object LCDNumber1:
TLCDNumber
Left = 0
Top = 0
Width = 237
Height = 35
Align = alTop
Color = clBackground
Digits = 9
ParentColor = False
SegmentStyle = ssFilled
Value = '00:00.000'
OnMouseUp =
LCDNumber1MouseUp
end
object Button1: TButton
Left = 2
Top = 40
Width = 75
Height = 25
Caption = 'Start' TabOrder = 1
OnClick = Button1Click
end
object Button2: TButton
Left = 80
Top = 40
Width = 75
Height = 25
Caption = 'Stop'
TabOrder = 2
OnClick = Button2Click
end
object Button3: TButton
Left = 158
Top = 40
<b>Width</b> = 75
Height = 25
Caption = 'Reset'
TabOrder = 3
OnClick = Button3Click
end
object Timer1: TTimer
Interval = 1
OnTimer = Timer1Timer
Left = 8
<b>Top</b> = 4
- P -

UpDown1: TUpDown;
procedure
FileListBox1Click(Sender:
TObject);
procedure ViewBtnClick(Sender:
TObject);
procedure ViewAsGlyph(const
FileExt: string);
procedure
GlyphCheckClick(Sender: TObject);
procedure
StretchCheckClick(Sender:
TObject);
procedure
FileEditKeyPress(Sender: TObject;
var Key: Char);
procedure
UpDownEditChange(Sender:
TObject);
procedure FormCreate(Sender:
TObject);
•
private
FormCaption: string;
end;
var
ImageForm: TImageForm;
implementation
uses ViewWin, SysUtils;
, N
{\$R *.dfm}
procedure
procedure TImageForm.FileListBox1Click(Sen
procedure
procedure TImageForm.FileListBox1Click(Sen der: TObject); var
procedure TImageForm.FileListBox1Click(Sen der: TObject);
procedure TImageForm.FileListBox1Click(Sen der: TObject); var
procedure TImageForm.FileListBox1Click(Sen der: TObject); var FileExt: string[4];
procedure TImageForm.FileListBox1Click(Sen der: TObject); var FileExt: string[4]; begin FileExt:=
procedure TImageForm.FileListBox1Click(Sen der: TObject); var FileExt: string[4]; begin FileExt:= AnsiUpperCase(ExtractFileExt(File
procedure TImageForm.FileListBox1Click(Sen der: TObject); var FileExt: string[4]; begin FileExt := AnsiUpperCase(ExtractFileExt(File ListBox1.Filename));
procedure TImageForm.FileListBox1Click(Sen der: TObject); var FileExt: string[4]; begin FileExt := AnsiUpperCase(ExtractFileExt(File ListBox1.Filename)); if (FileExt = '.BMP') or (FileExt =
procedure TImageForm.FileListBox1Click(Sen der: TObject); var FileExt: string[4]; begin FileExt := AnsiUpperCase(ExtractFileExt(File ListBox1.Filename)); if (FileExt = '.BMP') or (FileExt = '.ICO') or (FileExt = '.WMF') or
procedure TImageForm.FileListBox1Click(Sen der: TObject); var FileExt: string[4]; begin FileExt:= AnsiUpperCase(ExtractFileExt(File ListBox1.Filename)); if (FileExt = '.BMP') or (FileExt = '.ICO') or (FileExt = '.WMF') or (FileExt = '.EMF') then
procedure TImageForm.FileListBox1Click(Sen der: TObject); var FileExt: string[4]; begin FileExt := AnsiUpperCase(ExtractFileExt(File ListBox1.Filename)); if (FileExt = '.BMP') or (FileExt = '.ICO') or (FileExt = '.WMF') or
procedure TImageForm.FileListBox1Click(Sen der: TObject); var FileExt: string[4]; begin FileExt:= AnsiUpperCase(ExtractFileExt(File ListBox1.Filename)); if (FileExt = '.BMP') or (FileExt = '.ICO') or (FileExt = '.WMF') or (FileExt = '.EMF') then

الهلف التنفيذي لبرنامج				
(ImagView)				
عامدا				
علاء الدين اللباد				
هاتف ۱۷۳۵۷۵۶۹۰				
unit ImageWin;				
interface				
uses Windows, Classes, Graphics,				
Forms, Controls,				
FileCtrl, StdCtrls, ExtCtrls,				
Buttons, Spin, ComCtrls, Dialogs;				
type				
TImageForm = class(TForm)				
DirectoryListBox1:				
TDirectoryListBox;				
DriveComboBox1:				
TDriveComboBox;				
FileEdit: TEdit;				
UpDownGroup: TGroupBox;				
SpeedButton1: TSpeedButton;				
BitBtn1: TBitBtn;				
DisabledGrp: TGroupBox;				
SpeedButton2: TSpeedButton; BitBtn2: TBitBtn;				
Panel1: TPanel;				
Image1: TImage;				
FileListBox1: TFileListBox;				
Label2: TLabel;				
ViewBtn: TBitBtn;				
Bevel1: TBevel;				
Bevel2: TBevel;				
FilterComboBox1:				
TFilterComboBox;				
GlyphCheck: TCheckBox;				
StretchCheck: TCheckBox;				
UpDownEdit: TEdit;				

Image1.Picture.Bitmap;	istBox1.Filename);
SpeedButton2.Glyph :=	Caption := FormCaption +
Image1.Picture.Bitmap;	ExtractFilename(FileListBox1.Filen
UpDown1.Position :=	ame);
SpeedButton1.NumGlyphs;	if (FileExt = '.BMP') then
BitBtn1.Glyph :=	begin
Image1.Picture.Bitmap;	Caption := Caption +
BitBtn2.Glyph :=	Format(' (%d x %d)',
Image1.Picture.Bitmap;	[Image1.Picture.Width,
end	Image1. Picture. Height]);
else begin	ViewForm.Image1.Picture :=
SpeedButton1.Glyph := nil;	Image1.Picture;
SpeedButton2.Glyph := nil;	ViewForm.Caption := Caption;
BitBtn1.Glyph := nil;	if GlyphCheck.Checked then
BitBtn2.Glyph := nil;	ViewAsGlyph(FileExt);
end;	end
UpDown1.Enabled :=	else
GlyphCheck.Checked;	GlyphCheck.Checked := False;
UpDownEdit.Enabled :=	if FileExt = '.ICO' then
GlyphCheck.Checked;	begin
Label2.Enabled :=	Icon := Image1.Picture.Icon;
GlyphCheck.Checked;	ViewForm.Image1.Picture.Icon
end;	:= Icon;
enu,	end;
procedure	if (FileExt = '.WMF') or (FileExt =
TImageForm.ViewBtnClick(Sender:	'.EMF') then
TObject);	.E.vii ) then
begin	ViewForm.Image1.Picture.Metafile
ViewForm.HorzScrollBar.Range :=	:= Image1.Picture.Metafile;
Image1.Picture.Width;	end;
ViewForm.VertScrollBar.Range :=	end;
Image 1. Picture. Height;	chu,
ViewForm.Caption := Caption;	procedure
ViewForm.Show;	TImageForm.GlyphCheckClick(Sen
ViewForm.WindowState :=	der: TObject);
wsNormal;	begin
end;	v-Sin
CHU9	ViewAsGlyph(AnsiUpperCase(Extra
procedure	ctFileExt(FileListBox1.Filename)));
TImageForm.UpDownEditChange(S	end;
ender: TObject);	
resourcestring	procedure
sMinValue = 'Value below	TImageForm.ViewAsGlyph(const
minimum';	FileExt: string);
sMaxValue = 'Value over	begin
maximum';	if GlyphCheck.Checked and
,,	(FileExt = '.BMP') then
begin	begin
if (StrToInt(UpDownEdit.Text) <	SpeedButton1.Glyph :=
II (Sur round oppowing and read)	~pecabatton1.Gijpn .

end.	
///////////////////////////////////////	

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
Object8
الهك لبرناهم
(ImagView)
عداد
علاء الدين اللباد
واتف ۷۳۹۵۷۵۴۹۰
object ImageForm: TImageForm
Left = 193
Top = 107
ActiveControl = FileEdit
BorderIcons = [biSystemMenu,
biMinimize]
BorderStyle = bsSingle
Caption = 'Image Viewer'
ClientHeight = 315
Colon = all the Eagle
Color = clBtnFace ParentFont = True
OldCreateOrder = True
Position = poDefaultPosOnly
PixelsPerInch = 96
TextHeight = 13
object Bevel1: TBevel
Left = 316
Top = 5
Width = 171
Height = 121
end
object Bevel2: TBevel
Left = 317
Top = 132
Width = 170
Height = 177
end
object Label1: TLabel Left = 325
Top = 11
10p - 11 Width = 147
Height = 13
inight io

UpDown1.Min) then
begin
UpDownEdit.Text := '1';
raise
ERangeError.Create(sMinValue);
end
else if (StrToInt(UpDownEdit.Text)
> UpDown1.Max) then
begin
UpDownEdit.Text := '4';
raise
ERangeError.Create(sMaxValue);
end;
enu;
CwoodDuttou1 Nov. Cl 1
SpeedButton1.NumGlyphs :=
UpDown1.Position;
SpeedButton2.NumGlyphs :=
UpDown1.Position;
end;
procedure
TImageForm.StretchCheckClick(Se
nder: TObject);
begin
Image1.Stretch :=
StretchCheck.Checked;
end;
procedure
TImageForm.FileEditKeyPress(Sen
der: TObject; var Key: Char);
begin
if Key = #13 then
begin
FileListBox1.ApplyFilePath(FileEdit
.Text);
Key := #0;
end;
end;
procedure
TImageForm.FormCreate(Sender:
TObject);
begin
FormCaption := Caption + ' - ';
UpDown1.Min := 1;
UpDown1.Max := 4;
end;

Left = 116
Top = 26
Width = 25
Height = 25
AllowAllUp = True
GroupIndex = 1
end
object BitBtn1: TBitBtn
Left = 12
Top = 18
Width = 92
Height = 33
Caption = 'BitBtn1'
TabOrder = 0
end
end
object DisabledGrp: TGroupBox
Left = 323
Top = 220
Width = 154
Height = 60
Caption = 'Disabled'
TabOrder = 7
object SpeedButton2:
TSpeedButton
Left = 116
Top = 25 Width = 25
Height = 25
Enabled = False
end
object BitBtn2: TBitBtn
Left = 11
Top = 18
Width = 92
Height = 33
Caption = 'BitBtn2'
Enabled = False
TabOrder = 0
end
end
object Panel1: TPanel
Left = 324
Top = 27
Width = 153
Height = 68
BevelInner = bvLowered
TabOrder = 8
<u> </u>

AutoSize = False
Caption = 'Preview'
end
object Label2: TLabel
Left = 324
Top = 290
Width = 93
Height = 13
Caption = 'Number of Glyphs - '
end
object DirectoryListBox1:
TDirectoryListBox
Left = 8
Top = 12
Width = 148
Height = 260
FileList = FileListBox1
IntegralHeight = True
ItemHeight = 16
TabOrder = 1
end
object DriveComboBox1:
TDriveComboBox
Left = 9
Top = 277
Width = 148
Height = 19
DirList = DirectoryListBox1
TabOrder = 3
end
object FileEdit: TEdit
Left = 166
Top = 13
Width = 139
Height = 21
TabOrder = 0
Text = '*.bmp; *.ico; *.wmf; *.emf'
OnKeyPress = FileEditKeyPress
end
object UpDownGroup: TGroupBox
Left = 323
Top = 158
Width = 154
Height = 60
Caption = 'Up / Down'
TabOrder = 5
object SpeedButton1:
TSpeedButton

Top = 137
Width = 104
Height = 17
Caption = 'View as Glyph'
TabOrder = 9
OnClick = GlyphCheckClick
end
object StretchCheck: TCheckBox
Left = 413
Top = 96
Width = 64
Height = 17
Caption = 'Stretch'
TabOrder = 11
OnClick = StretchCheckClick
end
object UpDownEdit: TEdit
Left = 439
Top = 286
Width = 24
Height = 21
TabOrder = 10
Text = '1'
OnChange = UpDownEditChange
end
object UpDown1: TUpDown
Left = 463
Top = 286
Width = 11
Height = 21
Associate = UpDownEdit
Position = 1
Toshion = 1  TabOrder = 12
end
End
Did
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
object ViewForm: TViewForm
Left = 187
Top = 458
AutoScroll = False
Caption = 'View Form'
ClientHeight = 287
ClientWidth = 495
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET  Font Color = alWindowToxt
Font.Color = clWindowText

object Image1: TImage
Left = 2
Top = 2
Width = 149
Height = 64
Align = alClient
end
end
object FileListBox1: TFileListBox
Left = 166
Top = 41
Width = 139
Height = 227
FileEdit = FileEdit
ItemHeight = 13
Mask = '*.bmp; *.ico; *.wmf; *.emf'
TabOrder = 2
OnClick = FileListBox1Click
end
object ViewBtn: TBitBtn
Left = 325
Top = 98
Width = 63
Height = 24
Caption = '&Full View'
TabOrder = 6
OnClick = ViewBtnClick
end
object FilterComboBox1:
TFilterComboBox
Left = 166
Top = 275
Width = 140
Height = 21
FileList = FileListBox1
Filter =
'Image Files (*.bmp, *.ico,
*.wmf,
*.emf) *.bmp;*.ico;*.wmf;*.emf' +
' Bitmap Files
(*.bmp) *.bmp Icons
(*.ico) *.ico Metafiles (*.wmf' +
', *.emf) *.wmf;*.emf All files
(*.*) *.*'
TabOrder = 4
end
object GlyphCheck: TCheckBox
1 1
Left = 327

Left = 325
Top = 11
Width = 147
Height = 13
AutoSize = False
Caption = 'Preview'
end
object Label2: TLabel
Left = 324
Top = 290
Width = 93
Height = 13
Caption = 'Number of Glyphs - '
end
object DirectoryListBox1:
TDirectoryListBox
Left = 8
Top = 12
Width = 148
Height = 260
FileList = FileListBox1
IntegralHeight = True
ItemHeight = 16
TabOrder = 1
end
object DriveComboBox1:
TDriveComboBox
Left = 9
Top = 277
Width = 148
Height = 19
DirList = DirectoryListBox1
·
TabOrder = 3
object FileEdit: TEdit
Left = 166
Top = 13
Width = 139
Height = 21
TabOrder = 0
Text = '*.bmp;*.ico;*.wmf;*.emf'
OnKeyPress = FileEditKeyPress
end
object UpDownGroup: TGroupBox
Left = 323
Top = 158
Width = 154
Height = 60

Font.Name = 'System' Font.Style = [] OldCreateOrder = True Position = poDefault PixelsPerInch = 96 TextHeight = 16 object Image1: TImage Left = 0 Top = 0 Width = 495 Height = 287 Align = alClient AutoSize = True end End	
Font.Style = [] OldCreateOrder = True Position = poDefault PixelsPerInch = 96 TextHeight = 16 object Image1: TImage Left = 0 Top = 0 Width = 495 Height = 287 Align = alClient AutoSize = True end End  Object ImageForm: TImageForm Left = 193 Top = 107 ActiveControl = FileEdit BorderIcons = [biSystemMenu, biMinimize] BorderStyle = bsSingle Caption = 'Image Viewer' ClientHeight = 315 ClientWidth = 502 Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	Font.Height = -13
OldCreateOrder = True  Position = poDefault  PixelsPerInch = 96  TextHeight = 16  object Image1: TImage  Left = 0  Top = 0  Width = 495  Height = 287  Align = alClient  AutoSize = True  end  End  Object ImageForm: TImageForm  Left = 193  Top = 107  ActiveControl = FileEdit  BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle  Caption = 'Image Viewer'  ClientHeight = 315  ClientWidth = 502  Color = clBtnFace  ParentFont = True  OldCreateOrder = True  Position = poDefaultPosOnly  PixelsPerInch = 96  TextHeight = 13  object Bevel1: TBevel  Left = 316  Top = 5  Width = 171  Height = 121  end  object Bevel2: TBevel  Left = 317  Top = 132  Width = 170  Height = 177  end	Font.Name = 'System'
Position = poDefault PixelsPerInch = 96  TextHeight = 16 object Image1: TImage  Left = 0  Top = 0  Width = 495 Height = 287 Align = alClient AutoSize = True end  End  Object ImageForm: TImageForm  Left = 193  Top = 107  ActiveControl = FileEdit  BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle  Caption = 'Image Viewer'  ClientHeight = 315  ClientWidth = 502  Color = clBtnFace ParentFont = True  Position = poDefaultPosOnly PixelsPerInch = 96  TextHeight = 13 object Bevel1: TBevel  Left = 316  Top = 5  Width = 171  Height = 121 end object Bevel2: TBevel  Left = 317  Top = 132  Width = 170  Height = 177 end	Font.Style = []
PixelsPerInch = 96  TextHeight = 16  object Image1: TImage  Left = 0  Top = 0  Width = 495  Height = 287  Align = alClient  AutoSize = True end  End  Object ImageForm: TImageForm  Left = 193  Top = 107  ActiveControl = FileEdit  BorderStyle = bsSingle  Caption = 'Image Viewer'  ClientHeight = 315  ClientWidth = 502  Color = clBtnFace  ParentFont = True  Position = poDefaultPosOnly  PixelsPerInch = 96  TextHeight = 13  object Bevel1: TBevel  Left = 316  Top = 5  Width = 171  Height = 121 end  object Bevel2: TBevel  Left = 317  Top = 132  Width = 170  Height = 177 end	OldCreateOrder = True
TextHeight = 16 object Image1: TImage  Left = 0 Top = 0 Width = 495 Height = 287 Align = alClient AutoSize = True end End  Minimize BorderStyle = bsSingle Caption = 'Image Viewer' ClientHeight = 315 ClientWidth = 502 Color = clBtnFace ParentFont = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	Position = poDefault
object Image1: TImage  Left = 0  Top = 0  Width = 495  Height = 287  Align = alClient  AutoSize = True end  End  Object ImageForm: TImageForm  Left = 193  Top = 107  ActiveControl = FileEdit  BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle  Caption = 'Image Viewer'  ClientHeight = 315  ClientWidth = 502  Color = clBtnFace  ParentFont = True  OldCreateOrder = True  Position = poDefaultPosOnly  PixelsPerInch = 96  TextHeight = 13  object Bevel1: TBevel  Left = 316  Top = 5  Width = 171  Height = 121 end  object Bevel2: TBevel  Left = 317  Top = 132  Width = 170  Height = 177 end	PixelsPerInch = 96
Left = 0 Top = 0 Width = 495 Height = 287 Align = alClient AutoSize = True end End  Millimia  Object ImageForm: TImageForm Left = 193 Top = 107 ActiveControl = FileEdit BorderIcons = [biSystemMenu, biMinimize] BorderStyle = bsSingle Caption = 'Image Viewer' ClientHeight = 315 ClientWidth = 502 Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 Object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end Object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	TextHeight = 16
Left = 0 Top = 0 Width = 495 Height = 287 Align = alClient AutoSize = True end End  Millimia  Object ImageForm: TImageForm Left = 193 Top = 107 ActiveControl = FileEdit BorderIcons = [biSystemMenu, biMinimize] BorderStyle = bsSingle Caption = 'Image Viewer' ClientHeight = 315 ClientWidth = 502 Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 Object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end Object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	object Image1: TImage
Width = 495 Height = 287 Align = alClient AutoSize = True end End  Millimia  Object ImageForm: TImageForm Left = 193 Top = 107 ActiveControl = FileEdit BorderIcons = [biSystemMenu, biMinimize] BorderStyle = bsSingle Caption = 'Image Viewer' ClientHeight = 315 ClientWidth = 502 Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 Object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end Object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	
Height = 287 Align = alClient AutoSize = True end End  Willimia  object ImageForm: TImageForm Left = 193 Top = 107 ActiveControl = FileEdit BorderIcons = [biSystemMenu, biMinimize] BorderStyle = bsSingle Caption = 'Image Viewer' ClientHeight = 315 ClientWidth = 502 Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	Top = 0
Align = alClient AutoSize = True end End	•
Align = alClient AutoSize = True end End	Height = 287
AutoSize = True end End	<u> </u>
end End	
object ImageForm: TImageForm Left = 193 Top = 107 ActiveControl = FileEdit BorderIcons = [biSystemMenu, biMinimize] BorderStyle = bsSingle Caption = 'Image Viewer' ClientHeight = 315 ClientWidth = 502 Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	
object ImageForm: TImageForm  Left = 193  Top = 107  ActiveControl = FileEdit  BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle  Caption = 'Image Viewer'  ClientHeight = 315  ClientWidth = 502  Color = clBtnFace  ParentFont = True  OldCreateOrder = True  Position = poDefaultPosOnly  PixelsPerInch = 96  TextHeight = 13  object Bevel1: TBevel  Left = 316  Top = 5  Width = 171  Height = 121  end  object Bevel2: TBevel  Left = 317  Top = 132  Width = 170  Height = 177  end	
object ImageForm: TImageForm Left = 193 Top = 107 ActiveControl = FileEdit BorderIcons = [biSystemMenu, biMinimize] BorderStyle = bsSingle Caption = 'Image Viewer' ClientHeight = 315 ClientWidth = 502 Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	
object ImageForm: TImageForm Left = 193 Top = 107 ActiveControl = FileEdit BorderIcons = [biSystemMenu, biMinimize] BorderStyle = bsSingle Caption = 'Image Viewer' ClientHeight = 315 ClientWidth = 502 Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	\\\\\\\\
Top = 107 ActiveControl = FileEdit BorderIcons = [biSystemMenu, biMinimize] BorderStyle = bsSingle Caption = 'Image Viewer' ClientHeight = 315 ClientWidth = 502 Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	
Top = 107 ActiveControl = FileEdit BorderIcons = [biSystemMenu, biMinimize] BorderStyle = bsSingle Caption = 'Image Viewer' ClientHeight = 315 ClientWidth = 502 Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	ohiect ImageForm: TImageForm
Top = 107 ActiveControl = FileEdit BorderIcons = [biSystemMenu, biMinimize] BorderStyle = bsSingle Caption = 'Image Viewer' ClientHeight = 315 ClientWidth = 502 Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	
ActiveControl = FileEdit  BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle  Caption = 'Image Viewer'  ClientHeight = 315  ClientWidth = 502  Color = clBtnFace  ParentFont = True  OldCreateOrder = True  Position = poDefaultPosOnly  PixelsPerInch = 96  TextHeight = 13  object Bevel1: TBevel  Left = 316  Top = 5  Width = 171  Height = 121  end  object Bevel2: TBevel  Left = 317  Top = 132  Width = 170  Height = 177  end	
BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle  Caption = 'Image Viewer'  ClientHeight = 315  ClientWidth = 502  Color = clBtnFace  ParentFont = True  OldCreateOrder = True  Position = poDefaultPosOnly  PixelsPerInch = 96  TextHeight = 13  object Bevel1: TBevel  Left = 316  Top = 5  Width = 171  Height = 121  end  object Bevel2: TBevel  Left = 317  Top = 132  Width = 170  Height = 177  end	
biMinimize  BorderStyle = bsSingle Caption = 'Image Viewer' ClientHeight = 315 ClientWidth = 502 Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	
BorderStyle = bsSingle Caption = 'Image Viewer' ClientHeight = 315 ClientWidth = 502 Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	
Caption = 'Image Viewer' ClientHeight = 315 ClientWidth = 502 Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	
ClientHeight = 315 ClientWidth = 502 Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	
ClientWidth = 502 Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	
Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	
ParentFont = True OldCreateOrder = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	
OldCreateOrder = True Position = poDefaultPosOnly PixelsPerInch = 96 TextHeight = 13 object Bevel1: TBevel Left = 316 Top = 5 Width = 171 Height = 121 end object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	
Position = poDefaultPosOnly PixelsPerInch = 96  TextHeight = 13 object Bevel1: TBevel  Left = 316  Top = 5 Width = 171  Height = 121 end object Bevel2: TBevel  Left = 317  Top = 132 Width = 170  Height = 177 end	
PixelsPerInch = 96  TextHeight = 13  object Bevel1: TBevel  Left = 316  Top = 5  Width = 171  Height = 121  end  object Bevel2: TBevel  Left = 317  Top = 132  Width = 170  Height = 177  end	
TextHeight = 13  object Bevel1: TBevel  Left = 316  Top = 5  Width = 171  Height = 121  end  object Bevel2: TBevel  Left = 317  Top = 132  Width = 170  Height = 177  end	
object Bevel1: TBevel  Left = 316  Top = 5  Width = 171  Height = 121 end object Bevel2: TBevel  Left = 317  Top = 132  Width = 170  Height = 177 end	
Left = 316  Top = 5  Width = 171  Height = 121  end  object Bevel2: TBevel  Left = 317  Top = 132  Width = 170  Height = 177  end	
Top = 5 Width = 171 Height = 121 end object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	
Width = 171  Height = 121  end  object Bevel2: TBevel  Left = 317  Top = 132  Width = 170  Height = 177  end	
Height = 121 end object Bevel2: TBevel  Left = 317 Top = 132 Width = 170 Height = 177 end	•
end object Bevel2: TBevel Left = 317 Top = 132 Width = 170 Height = 177 end	
object Bevel2: TBevel  Left = 317  Top = 132  Width = 170  Height = 177  end	
Left = 317 Top = 132 Width = 170 Height = 177 end	
Top = 132 Width = 170 Height = 177 end	- v
Width = 170 Height = 177 end	
Height = 177 end	•
end	
object Labell, TLabel	
onject Labett: Thabet	object Label1: TLabel

Width = 153
Height = 68
BevelInner = bvLowered
TabOrder = 8
object Image1: TImage
Left = 2
Top = 2
Width = 149
Height = 64
Align = alClient
end
end
object FileListBox1: TFileListBox
Left = 166
Top = 41
Width = 139
Height = 227
FileEdit = FileEdit
ItemHeight = 13
Mask = '*.bmp; *.ico; *.wmf; *.emf'
TabOrder = 2
OnClick = FileListBox1Click
end
object ViewBtn: TBitBtn
Left = 325
Top = 98
Width = 63
Height = 24
Caption = '&Full View'
TabOrder = 6
OnClick = ViewBtnClick
end
object FilterComboBox1:
TFilterComboBox
Left = 166
Top = 275
Width = 140
Height = 21
FileList = FileListBox1
Filter =
'Image Files (*.bmp, *.ico,
*.wmf,
*.emf) *.bmp;*.ico;*.wmf;*.emf' +
' Bitmap Files
(*.bmp) *.bmp Icons
(*.ico) *.ico Metafiles (*.wmf' +
', *.emf) *.wmf;*.emf All files
(*.*) *.*'

Caption = 'Up / Down'
TabOrder = 5
object SpeedButton1:
TSpeedButton
Left = 116
Top = 26
Width = 25
Height = 25
AllowAllUp = True
GroupIndex = 1
end
object BitBtn1: TBitBtn
Left = 12
Top = 18
Width = 92
Height = 33
Caption = 'BitBtn1'
TabOrder = 0
end
end
object DisabledGrp: TGroupBox
Left = 323
Top = 220
Width = 154
Height = 60
Caption = 'Disabled'
TabOrder = 7
object SpeedButton2:
TSpeedButton
Left = 116
Top = 25
Width = 25
Height = 25
Enabled = False
end
object BitBtn2: TBitBtn
Left = 11
Top = 18
Width = 92
Height = 33
Caption = 'BitBtn2'
Enabled = False
TabOrder = 0
end
end
object Panel1: TPanel
Left = 324
Top = 27

TabOrder = 4
end
object GlyphCheck: TCheckBox
Left = 327
Top = 137
Width = 104
Height = 17
Caption = 'View as Glyph'
TabOrder = 9
OnClick = GlyphCheckClick
end
object StretchCheck: TCheckBox
Left = 413
Top = 96
Width = 64
Height = 17
Caption = 'Stretch'
TabOrder = 11
OnClick = StretchCheckClick
end
object UpDownEdit: TEdit
Left = 439
Top = 286
Width = 24
Height = 21
TabOrder = 10
Text = '1'
OnChange = UpDownEditChange
end
object UpDown1: TUpDown
Left = 463
Top = 286
Width = 11
Height = 21
Associate = UpDownEdit
Position = 1
TabOrder = 12
end
End
111111111111111111111111111111111111111

TObject);
private
ThreadsRunning: Integer;
procedure RandomizeArrays;
procedure ThreadDone(Sender:
TObject);
public
procedure PaintArray(Box:
TPaintBox; const A: array of
Integer);
end;
var
ThreadSortForm:
TThreadSortForm;
implementation
uses SortThds;
{\$R *.dfm}
type
PSortArray = ^TSortArray;
TSortArray = array[0114] of
Integer;
var
ArraysRandom: Boolean;
BubbleSortArray,
SelectionSortArray,
QuickSortArray: TSortArray;
{ TThreadSortForm }
procedure
TThreadSortForm.PaintArray(Box:
TPaintBox; const A: array of
Integer);
var
I: Integer;
begin
with Box do
begin
Canvas.Pen.Color := clRed;
for I := Low(A) to High(A) do
PaintLine(Canvas, I, A[I]);
end;
end;
,

الهلف لبرناهم
(Threads)
عامدا
علاء الدين اللباد
واتف ۷۳۷۱۹۹۱
unit ThSort;
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
ExtCtrls, StdCtrls;
type
TThreadSortForm = class(TForm)
StartBtn: TButton;
BubbleSortBox: TPaintBox;
SelectionSortBox: TPaintBox;
QuickSortBox: TPaintBox;
Label1: TLabel;
Bevel1: TBevel;
Bevel2: TBevel;
Bevel3: TBevel;
Label2: TLabel;
Label3: TLabel;
procedure
BubbleSortBoxPaint(Sender:
TObject);
procedure SelectionSortBoxPaint(Sender:
TObject);
procedure
QuickSortBoxPaint(Sender:
TObject);
procedure FormCreate(Sender:
TObject);
procedure StartBtnClick(Sender:

0.77
OnTerminate := ThreadDone;
StartBtn.Enabled := False;
end;
procedure
TThreadSortForm.RandomizeArray
s;
var
I: Integer;
begin
if not ArraysRandom then
begin
Randomize;
for I := Low(BubbleSortArray) to
High(BubbleSortArray) do
BubbleSortArray[I] :=
Random(170);
SelectionSortArray :=
BubbleSortArray;
QuickSortArray :=
BubbleSortArray;
ArraysRandom := True;
Repaint;
end;
end;
,
procedure
TThreadSortForm.ThreadDone(Sen
der: TObject);
begin
Dec(ThreadsRunning);
if ThreadsRunning = 0 then
begin
StartBtn.Enabled := True;
ArraysRandom := False;
end;
end;
Ciiu,
and
end.
111111111111111111111111111111111111111
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

procedure
TThreadSortForm.BubbleSortBoxP
aint(Sender: TObject);
begin
PaintArray(BubbleSortBox,
BubbleSortArray);
end;
procedure
TThreadSortForm.SelectionSortBox
Paint(Sender: TObject);
Beint A may (Selection Sout Day
PaintArray(SelectionSortBox, SelectionSortArray);
end;
CHU,
procedure
TThreadSortForm.QuickSortBoxPai
nt(Sender: TObject);
begin
PaintArray(QuickSortBox,
QuickSortArray);
end;
,
procedure
TThreadSortForm.FormCreate(Sen
der: TObject);
begin
RandomizeArrays;
end;
procedure
TThreadSortForm.StartBtnClick(Se
nder: TObject);
Bandamiza A mayer
RandomizeArrays;
ThreadsRunning := 3; with
TBubbleSort.Create(BubbleSortBox
, BubbleSortArray) do
OnTerminate := ThreadDone;
with
TSelectionSort.Create(SelectionSort
Box, SelectionSortArray) do
OnTerminate := ThreadDone;
with
TQuickSort.Create(QuickSortBox,
QuickSortArray) do

Left = 192
Top = 24
Width = 177
Height = 233
end
object BubbleSortBox: TPaintBox
Left = 8
Top = 24
Width = 177
Height = 233
OnPaint = BubbleSortBoxPaint
end
object SelectionSortBox:
TPaintBox
Left = 192
Top = 24
Width = 177
Height = 233
OnPaint = SelectionSortBoxPaint
end
object QuickSortBox: TPaintBox
Left = 376
Top = 24
Width = 177
Height = 233
OnPaint = QuickSortBoxPaint
end
object Label1: TLabel
Left = 8
Top = 8
Width = 55
Height = 13
Caption = 'Bubble Sort'
end
object Label2: TLabel
Left = 192
Top = 8
Width = 66
Height = 13
Caption = 'Selection Sort'
end
object Label3: TLabel
Left = 376
Top = 8
Width = 50
Height = 13
Caption = 'Quick Sort'
end

ObjectS
الملف لبرنامج
(Threads)
عامدا
علاء الدين اللباد
هاتف ۹۶۵۷۵۳۷۱
object ThreadSortForm: TThreadSortForm
Left = 182
Top = 115
BorderStyle = bsDialog
Caption = 'Thread Sorting Demo'
ClientHeight = 295
ClientWidth = 562
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
object Bevel1: TBevel
Left = 8
Top = 24
Width = 177
Height = 233
end
object Bevel3: TBevel
Left = 376
Top = 24
Width = 177 Height = 233
end
object Bevel2: TBevel
unjett beveiz. I bevei

public
{ Public declarations }
end;
var
Form1: TForm1;
implementation
{\$R *.dfm}
(SK .um)
<> procedure
TForm1.FormCreate(Sender:
TObject);
begin
Listbox1.Items := Screen.Fonts;
end;
<> procedure
TForm1.ListBox1Click(Sender:
TObject);
begin
FontLabel.Caption :=
ListBox1.Items[ListBox1.ItemIndex]
end;
<> procedure
TForm1.DrawItem(Control:
TWinControl; Index: Integer; Rect: TRect; State:
TOwnerDrawState);
begin
with ListBox1.Canvas do
begin
FillRect(Rect);
Font.Name := ListBox1.Items[Index];
Font.Size := 0; // use font's
preferred size
TextOut(Rect.Left+1, Rect.Top+1,
ListBox1.Items[Index]);
end;
end;
<> procedure
TForm1.ListBox1MeasureItem(Cont
rol: TWinControl; Index: Integer;
Tot: I Wincontrol, Index: Integer,

object StartBtn: TButton
Left = 480
Top = 264
Width = 75
Height = 25
Caption = 'Start Sorting'
TabOrder = 0
OnClick = StartBtnClick
end
end

الهلف التنفيذي لبرناهج
8 procedures
Ownerlst)
عامدا
علاء الدين اللباد
هاتف ۷۹۲۵۷۵۳۷۱
unit Fontlist;
intonfogo
interface
uses Windows, Classes, Graphics,
Forms, Controls, StdCtrls;
type
TForm1 = class(TForm)
ListBox1: TListBox;
Label1: TLabel;
FontLabel: TLabel;
<> procedure
FormCreate(Sender: TObject);
<> procedure
ListBox1Click(Sender: TObject);
<> procedure
DrawItem(Control: TWinControl;
Index: Integer; Rect: TRect;
State: TOwnerDrawState);
<> procedure
ListBox1MeasureItem(Control:
TWinControl; Index: Integer;
var Height: Integer);
private { Private declarations }
{ I Tivate ucciai ations }

Height = 16
Caption = 'System Fonts'
end
<> object FontLabel: TLabel
Left = 20
Top = 305
Width = 241
Height = 20
AutoSize = False
end
<> object ListBox1: TListBox
Left = 20
Top = 45
Width = 241
Height = 251
Style = lbOwnerDrawVariable
ItemHeight = 16
TabOrder = 0
OnClick = ListBox1Click
OnDrawItem = DrawItem
OnMeasureItem =
ListBox1MeasureItem
end
end

begin
with ListBox1.Canvas do
begin
Font.Name :=
Listbox1.Items[Index];
Font.Size := 0; // use
font's preferred size
<pre>Height := TextHeight('Wg') + 2; //</pre>
measure ascenders and descenders
end;
end;
end.

<----> objects4

الملف النصي لبرنامج

( Owner ist)
إعداد
علاء الدين اللباد
واتف ۷۳۹۵۷۵۹۹۰
<> object Form1: TForm1
Left = 174
Top = 108
ActiveControl = ListBox1
BorderStyle = bsDialog
Caption = 'OwnerDraw'
ClientHeight = 359
ClientWidth = 276
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Scaled = False
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 16
<> object Label1: TLabel
Left = 20
Top = 20
Width = 81
Width = 81

StopBtn: TToolButton;
RefreshBtn: TToolButton;
ToolBar2: TToolBar;
ToolButton6: TToolButton;
ToolButton5: TToolButton;
ToolButton7: TToolButton;
ToolButton8: TToolButton;
Animate1: TAnimate;
URLs: TComboBox;
,
Help1: TMenuItem;
About1: TMenuItem;
Toolbar3: TMenuItem;
Statusbar2: TMenuItem;
Go1: TMenuItem;
Back1: TMenuItem;
Forward1: TMenuItem;
Stop1: TMenuItem;
Refresh1: TMenuItem;
N2: TMenuItem;
ActionList1: TActionList;
BackAction: TAction;
ForwardAction: TAction;
StopAction: TAction;
RefreshAction: TAction;
WebBrowser1: TWebBrowser;
<> procedure
Exit1Click(Sender: TObject);
<> procedure
About1Click(Sender: TObject);
<> procedure
StopClick(Sender: TObject);
<> procedure
URLsKeyDown(Sender: TObject;
var Key: Word;
Shift: TShiftState);
<> procedure
FormCreate(Sender: TObject);
<> procedure
LinksClick(Sender: TObject);
<> procedure
RefreshClick(Sender: TObject);
<> procedure
BackClick(Sender: TObject);
<> procedure
ForwardClick(Sender: TObject);
<> procedure Form Destroy(Sendon, TObject)
FormDestroy(Sender: TObject);
<> procedure
URLsClick(Sender: TObject);

الهاف التنفيذي لبرناهم
procedure50
(CoolStuf)
عامدا
علاء الدين اللباد
۶۹۳۳ ۱۸۵۸۵۶۶۰۰ ۱
unit Main;
•
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
StdCtrls, ExtCtrls, Menus,
ComCtrls, OleCtrls, Buttons,
ToolWin, Isp3,
ActnList, ImgList, SHDocVw;
const
CM_HOMEPAGEREQUEST =
WM_USER + \$1000;
type
TMainForm = class(TForm)
StatusBar1: TStatusBar;
MainMenu1: TMainMenu;
File1: TMenuItem;
Exit1: TMenuItem;
View1: TMenuItem;
NavigatorImages: TImageList;
NavigatorHotImages:
TImageList;
LinksImages: TImageList;
LinksHotImages: TImageList;
CoolBar1: TCoolBar;
ToolBar1: TToolBar;
BackBtn: TToolButton;
ForwardBtn: TToolButton;

TMainForm.Exit1Click(Sender:	<> procedure
TObject);	FormKeyDown(Sender: TObject;
begin	var Key: Word;
Close;	Shift: TShiftState);
end;	<> procedure
	Toolbar3Click(Sender: TObject);
<> procedure	<> procedure
TMainForm.FindAddress;	Statusbar2Click(Sender: TObject);
var	<> procedure
Flags: OLEVariant;	BackActionUpdate(Sender:
	TObject);
begin	<> procedure
Flags := 0;	ForwardActionUpdate(Sender:
UpdateCombo := True;	TObject);
	<> procedure
WebBrowser1.Navigate(WideString(	WebBrowser1BeforeNavigate2(Send
Urls.Text), Flags, Flags, Flags,	er: TObject;
Flags);	const pDisp: IDispatch; var
end;	URL, Flags, TargetFrameName,
,	PostData,
<> procedure	Headers: OleVariant; var
TMainForm.About1Click(Sender:	Cancel: WordBool);
TObject);	<> procedure
begin	WebBrowser1DownloadBegin(Sende
ShowAboutBox;	r: TObject);
end;	<> procedure
	WebBrowser1DownloadComplete(S
<> procedure	ender: TObject);
TMainForm.StopClick(Sender:	private
TObject);	HistoryIndex: Integer;
begin	HistoryList: TStringList;
WebBrowser1.Stop;	UpdateCombo: Boolean;
end;	<> procedure FindAddress;
Circi,	<> procedure
<> procedure	HomePageRequest(var message:
TMainForm.URLsKeyDown(Sender	tmessage); message
: TObject; var Key: Word;	CM_HOMEPAGEREQUEST;
Shift: TShiftState);	end;
begin	
if Key = VK Return then	var
begin	MainForm: TMainForm;
FindAddress;	,
end;	implementation
-	•
end;	uses About;
z > nwoooduwo	
<> procedure	{\$R *.dfm}
TMainForm.URLsClick(Sender:	(*** ******)
TObject);	<> procedure
begin	proccuure

BackBtn.Enabled then	FindAddress;
BackBtn.Click;	end;
end;	
	<> procedure
<> procedure	TMainForm.LinksClick(Sender:
TMainForm.Toolbar3Click(Sender:	TObject);
TObject);	begin
begin	if (Sender as TToolButton).Hint = "
with Sender as TMenuItem do	then Exit;
begin	URLs.Text := (Sender as
Checked := not Checked;	TToolButton).Hint;
Coolbar1.Visible := Checked;	FindAddress;
end;	end;
end;	·
	<> procedure
<> procedure	TMainForm.RefreshClick(Sender:
TMainForm.Statusbar2Click(Sende	TObject);
r: TObject);	begin
begin	FindAddress;
with Sender as TMenuItem do	end;
	cnu,
begin Charles I. and Charles I.	< > nwaaaduwa
Checked := not Checked;	<pre>&lt;&gt; procedure TMoinForm PoolsClick(Sondon)</pre>
StatusBar1.Visible := Checked;	TMainForm.BackClick(Sender:
end;	TObject);
end;	begin
	URLs.Text :=
<> procedure	HistoryList[HistoryIndex - 1];
TMainForm.HomePageRequest(var	FindAddress;
Message: TMessage);	end;
begin	
URLs.Text :=	<> procedure
'http://www.borland.com';	TMainForm.ForwardClick(Sender:
FindAddress;	TObject);
end;	begin
	URLs.Text :=
<> procedure	HistoryList[HistoryIndex + 1];
TMainForm.FormCreate(Sender:	FindAddress;
TObject);	end;
begin	
HistoryIndex := -1;	<> procedure
HistoryList := TStringList.Create;	TMainForm.FormKeyDown(Sender
{ Load the animation from the AVI	: TObject; var Key: Word;
file in the startup directory. An	Shift: TShiftState);
alternative to this would be to	begin
create a .RES file including the	if Shift = [ssAlt] then
S	if (Key = VK RIGHT) and
cool.avi	ForwardBtn.Enabled then
as an AVI resource and use the	ForwardBtn.Click
ResName or ResId properties of	rorwarudui.Click
Animate1 to	else if (Key = VK LEFT) and

	main4 (a i4 )
Name Indiana Indiana	point to it. }
NewIndex: Integer;	Animate1.FileName :=
begin	ExtractFilePath(Application.ExeNa
NewIndex :=	me) + 'cool.avi';
HistoryList.IndexOf(URL);	{ Find the home page - needs to be
if NewIndex = -1 then	posted because HTML control
begin	hasn't been
{ Remove entries in HistoryList	registered yet. }
between last address and current	PostMessage(Handle,
address }	CM_HOMEPAGEREQUEST, 0, 0);
if (HistoryIndex >= 0) and	end;
(HistoryIndex < HistoryList.Count -	
1) then	<> procedure
while HistoryList.Count >	TMainForm.FormDestroy(Sender:
HistoryIndex do	TObject);
-	begin
HistoryList.Delete(HistoryIndex);	HistoryList.Free;
HistoryIndex :=	end;
HistoryList.Add(URL);	,
end	<> procedure
else	TMainForm.BackActionUpdate(Sen
HistoryIndex := NewIndex;	der: TObject);
if UpdateCombo then	begin
begin	if HistoryList.Count > 0 then
UpdateCombo := False;	BackAction.Enabled :=
NewIndex :=	
	HistoryIndex > 0 else
URLs.Items.IndexOf(URL); if NewIndex = -1 then	BackAction.Enabled := False;
	,
URLs.Items.Insert(0, URL)	end;
else	
URLs.Items.Move(NewIndex, 0);	<> procedure
end;	TMainForm.ForwardActionUpdate(
URLs.Text := URL;	Sender: TObject);
Statusbar1.Panels[0].Text := URL;	begin
end;	if HistoryList.Count > 0 then
	ForwardAction.Enabled :=
<> procedure	HistoryIndex < HistoryList.Count -
TMainForm.WebBrowser1Downloa	1
dBegin(Sender: TObject);	else
begin	ForwardAction.Enabled := False;
{ Turn the stop button dark red }	end;
StopBtn.ImageIndex := 4;	
{ Play the avi from the first frame	<> procedure
indefinitely }	TMainForm.WebBrowser1BeforeNa
Animate1.Active := True;	vigate2(Sender: TObject;
end;	const pDisp: IDispatch; var URL,
	Flags, TargetFrameName, PostData,
<> procedure	Headers: OleVariant; var Cancel:
TMainForm.WebBrowser1Downloa	WordBool);
1 141 amil of m. 44 codi om sci i Dominos	

<> procedure ShowAboutBox;
begin
with
TAboutBox.Create(Application) do
try
ShowModal;
finally
Free;
end;
end;
-
<> procedure
TAboutBox.GetOSInfo;
var
Platform: string;
BuildNumber: Integer;
begin Cappy of
case Win32Platform of
AMED DI AMEADAN MUNICA MUNICA
VER_PLATFORM_WIN32_WIND
OWS:
begin
Platform := 'Windows 95';
BuildNumber := Win32BuildNumber and
\$0000FFFF; end;
VER PLATFORM WIN32 NT:
begin Platform := 'Windows NT';
BuildNumber :=
Win32BuildNumber; end;
elu; else
begin
Platform := 'Windows'; BuildNumber := 0;
end;
,
end;
if (Win32Platform = VER_PLATFORM_WIN32_WIND
OWS) or
(Win32Platform =
VER PLATFORM WIN32 NT)
then
begin
if Win32CSDVersion = " then
OS.Caption := Format('%s
%d.%d (Build %d)', [Platform,
/vu./vu (Dunu /vu) , [i lativi iii,

dComplete(Sender: TObject);
begin
{ Turn the stop button grey }
StopBtn.ImageIndex := 2;
{ Stop the avi and show the first
, <u> </u>
frame }
Animate1.Active := False;
end;
end.
\\\\\\\
unit About;
interface
uses
SysUtils, Windows, Messages,
Classes, Graphics, Controls,
Forms, Dialogs, StdCtrls, Buttons,
ExtCtrls;
type
TAboutBox = class(TForm)
OKButton: TButton;
Copyright: TLabel;
Bevel1: TBevel;
,
SKUName: TLabel;
Image2: TImage;
PhysMem: TLabel;
OS: TLabel;
Label3: TLabel;
Image1: TImage;
<> procedure
FormCreate(Sender: TObject);
private
<> procedure GetOSInfo;
<> procedure
InitializeCaptions;
end;
)
/ > nyogoduwa Chaw A hautDawa
<> procedure ShowAboutBox;
implementation
uses ShellAPI;
,
{\$R *.dfm}
(wit min)

علاء الدين اللباد
واتة ۷۳۹۵۷۵۶۹۰
<> object MainForm:
TMainForm Left = 173
Top = 148
Width = 503
Height = 370
Caption = 'Web Browser' Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11 Font.Name = 'MS Sans Serif'
Font.Style = []
Menu = MainMenu1
OldCreateOrder = True
Position = poScreenCenter Scaled = False
ShowHint = True
Visible = True
OnCreate = FormCreate
OnDestroy = FormDestroy OnKeyDown = FormKeyDown
PixelsPerInch = 96
TextHeight = 13
<> object StatusBar1:
TStatusBar Left = 0
Top = 297
Width = 495
Height = 19
Panels = < item
Width = 50
end>
end
<> object CoolBar1: TCoolBar
Left = 0
Top = 0
Width = 495
Height = 91 AutoSize = True
Bands = <

Win32MajorVersion, Win32MinorVersion, BuildNumber])
BuildNumber])
else
OS.Caption := Format('%s
%d.%d (Build %d: %s)', [Platform,
Win32MajorVersion,
Win32MinorVersion,
BuildNumber, Win32CSDVersion]);
end
else
OS.Caption := Format('%s
%d.%d', [Platform,
Win32MajorVersion,
Win32MinorVersion])
end;
<> procedure
TAboutBox.InitializeCaptions;
var
MS: TMemoryStatus;
begin
GetOSInfo;
MS.dwLength :=
SizeOf(TMemoryStatus);
GlobalMemoryStatus(MS);
PhysMem.Caption :=
FormatFloat('#,###" KB"',
MS.dwTotalPhys div 1024);
end;
<> procedure
TAboutBox.FormCreate(Sender:
TObject);
begin
InitializeCaptions;
with Image2.Picture.Bitmap do
Handle :=
CreateGrayMappedBmp(Handle);
end;
end.

Objects 46	
الهلف النصي لبرناهج	
(CoolStuf)	
عامدا	

ShowCaptions = True
TabOrder = 0
Transparent = True
Wrapable = False
<> object BackBtn:
TToolButton
Left = 0
Top = 0
Action = BackAction
end
<> object ForwardBtn:
TToolButton
Left = 45
Top = 0
Action = ForwardAction
end
<> object StopBtn:
TToolButton
Left = 90
Top = 0
Action = StopAction
end
<> object RefreshBtn:
TToolButton
Left = 135
Top = 0
Action = RefreshAction
end
end
<> object ToolBar2:
TToolBar
Left = 38
Top = 65
Width = 449
Height = 22
AutoSize = True
ButtonWidth = 94
EdgeBorders = []
Flat = True
HotImages = LinksHotImages
Images = LinksImages
List = True
ShowCaptions = True
TabOrder = 1
Transparent = True
Wrapable = False
<> object ToolButton6:
TToolButton

item
Break = False
Control = ToolBar1
ImageIndex = -1
MinHeight = 40
Width = 429
end
item
Break = False
Control = Animate1
FixedSize = True
ImageIndex = -1
MinHeight = 39
MinWidth = 60
Width = 60
end
item
Control = URLs
ImageIndex = -1
MinHeight = 21
Text = 'Address'
Width = 491
end
item
Control = ToolBar2
ImageIndex = -1
MinHeight = 22
Text = 'Links'
Width = 491
end>
ParentShowHint = False
Bitmap.Data = {
}
ShowHint = False
<> object ToolBar1:
TToolBar
Left = 9
Top = 0
Width = 416
Height = 40
AutoSize = True
ButtonHeight = 40
ButtonWidth = 45
EdgeBorders = []
Flat = True
HotImages =
NavigatorHotImages
Images = NavigatorImages

end
<> object URLs:
TComboBox
Left = 51
Top = 42
Width = 436
Height = 21
Hint = 'URL'
ItemHeight = 13
TabOrder = 3
OnClick = URLsClick
OnKeyDown = URLsKeyDown
end
end
<pre>&lt;&gt; object WebBrowser1:</pre>
TWebBrowser
Left = 0
Top = 91
Width = 495
Height = 206
Align = alClient
TabOrder = 2
OnDownloadBegin =
WebBrowser1DownloadBegin
OnDownloadComplete =
WebBrowser1DownloadComplete
OnBeforeNavigate2 =
WebBrowser1BeforeNavigate2
ControlData = {
}
end
<> object MainMenu1:
TMainMenu
Left = 128
Top = 200
<> object File1: TMenuItem
Caption = '&File'
<> object Exit1:
TMenuItem
Caption = 'E&xit'
OnClick = Exit1Click
end
end
<> object View1:
TMenuItem
Caption = '&View'
<> object Toolbar3:
TMenuItem

7.0.0
Left = 0
Top = 0
Hint =
'http://www.informant.com:80/delph
i/'
Caption = 'Informant'
ImageIndex = 0
OnClick = LinksClick
end
<> object ToolButton5:
TToolButton
Left = 94
Top = 0
Hint = 'http://www.delphi-
jedi.org/'
Caption = 'Delphi Jedi'
ImageIndex = 1
OnClick = LinksClick
end
<> object ToolButton7:
TToolButton
Left = 188
Top = 0
Hint =
'http://www.borland.com:80/delphi'
Caption = 'Product News'
ImageIndex = 2
OnClick = LinksClick
end
<> object ToolButton8:
TToolButton
Left = 282
Top = 0
Hint =
'http://www.borland.com:80'
-
Caption = 'Borland'
ImageIndex = 3
OnClick = LinksClick
end
end
<> object Animate1:
TAnimate
Left = 431
Top = 0
Width = 60
Height = 39
FileName = 'Cool.avi'
StopFrame = 13

Top = 104
Bitmap = {
end
<> object ActionList1:
TActionList
Images = NavigatorImages
Left = 80
Top = 160
<> object BackAction:
TAction
Caption = 'Back'
ImageIndex = 0
OnExecute = BackClick
OnUpdate = BackActionUpdate
end
<> object ForwardAction:
TAction
Caption = 'Forward'
Enabled = False
ImageIndex = 1
OnExecute = ForwardClick
OnUpdate =
ForwardActionUpdate
end
<> object StopAction:
TAction
Caption = 'Stop'
ImageIndex = 2
OnExecute = StopClick
end
<> object RefreshAction:
TAction
Caption = 'Refresh'
ImageIndex = 3
OnExecute = RefreshClick
end
end
End
<> object AboutBox:
TAboutBox
Left = 142
Top = 116
BorderStyle = bsDialog
Caption = 'About Web Browser'
ClientHeight = 258 ClientWidth = 342
Chefft With - 342

Caption = '&Toolbar'
Checked = True
OnClick = Toolbar3Click
end
<> object Statusbar2:
TMenuItem
Caption = 'Status &Bar'
Checked = True
OnClick = Statusbar2Click
end
<> object N2: TMenuItem
Caption = '-'
end
<> object Stop1:
TMenuItem
Action = StopAction
end
<> object Refresh1:
TMenuItem
Action = RefreshAction
end
end
<> object Go1: TMenuItem
Caption = '&Go'
<> object Back1:
TMenuItem
Action = BackAction
end
<> object Forward1:
TMenuItem
Action = ForwardAction
end
end
<> object Help1:
TMenuItem
Caption = '&Help'
<> object About1:
TMenuItem
Caption = '&About'
OnClick = About1Click
end
end
end
<> object NavigatorImages:
TImageList
Height = 20
Width = 30
Left = 32

end <> object Label3: TLabel Left = 86 Top = 184 Width = 145 Height = 13 Caption = 'Memory Available to Windows:' end
Left = 86 Top = 184 Width = 145 Height = 13 Caption = 'Memory Available to Windows:'
Left = 86 Top = 184 Width = 145 Height = 13 Caption = 'Memory Available to Windows:'
Width = 145 Height = 13 Caption = 'Memory Available to Windows:'
Height = 13 Caption = 'Memory Available to Windows:'
Caption = 'Memory Available to Windows:'
Windows:'
_
end
<> object PhysMem: TLabel
Left = 323
Top = 184
Width = 6
Height = 13
Alignment = taRightJustify
Caption = '0'
end
<> object OS: TLabel
Left = 86
Top = 167
Width = 15
Height = 13
Caption = 'OS'
end
<> object Image1: TImage
Left = 9
Top = 11
Width = 60
Height = 39
AutoSize = True
Picture.Data = {
0000}
Transparent = True
end
<> object OKButton: TButton
Left = 256
Top = 223
Width = 75
Height = 25
Cancel = True
Caption = 'OK'
Default = True
ModalResult = 1
TabOrder = 0
end
end

Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Copyright: TLabel Left = 82 Top = 81 Width = 207 Height = 13 Caption = 'Copyright 1983 - 2002 Borland International' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style =    ParentFont = False end <> object Bevel1: TBevel Left = 86 Top = 158 Width = 251 Height = 2 end <> object SKUName: TLabel Left = 82 Top = 57 Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style =    ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = { 0003	
OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Copyright: TLabel Left = 82 Top = 81 Width = 207 Height = 13 Caption = 'Copyright 1983 - 2002 Borland International' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Bevel1: TBevel Left = 86 Top = 158 Width = 251 Height = 2 end <> object SKUName: TLabel Left = 82 Top = 57 Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	Color = clBtnFace
Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Copyright: TLabel Left = 82 Top = 81 Width = 207 Height = 13 Caption = 'Copyright 1983 - 2002 Borland International' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Bevel1: TBevel Left = 86 Top = 158 Width = 251 Height = 2 end <> object SKUName: TLabel Left = 82 Top = 57 Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	ParentFont = True
OnCreate = FormCreate  PixelsPerInch = 96  TextHeight = 13  <> object Copyright: TLabel  Left = 82  Top = 81  Width = 207  Height = 13  Caption = 'Copyright 1983 - 2002  Borland International'  Font.Charset = ANSI_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  ParentFont = False end  <> object Bevel1: TBevel  Left = 86  Top = 158  Width = 251  Height = 2 end  <> object SKUName: TLabel  Left = 82  Top = 57  Width = 97  Height = 16  Caption = 'Using Delphi 7.0'  Font.Charset = ANSI_CHARSET  Font.Color = clBlack  Font.Height = -13  Font.Name = 'MS Sans Serif'  Font.Style = []  ParentFont = False end  <> object Image2: TImage  Left = 76  Top = 13  Width = 170  Height = 35  AutoSize = True  Picture.Data = {	OldCreateOrder = True
OnCreate = FormCreate  PixelsPerInch = 96  TextHeight = 13  <> object Copyright: TLabel  Left = 82  Top = 81  Width = 207  Height = 13  Caption = 'Copyright 1983 - 2002  Borland International'  Font.Charset = ANSI_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  ParentFont = False end  <> object Bevel1: TBevel  Left = 86  Top = 158  Width = 251  Height = 2 end  <> object SKUName: TLabel  Left = 82  Top = 57  Width = 97  Height = 16  Caption = 'Using Delphi 7.0'  Font.Charset = ANSI_CHARSET  Font.Color = clBlack  Font.Height = -13  Font.Name = 'MS Sans Serif'  Font.Style = []  ParentFont = False end  <> object Image2: TImage  Left = 76  Top = 13  Width = 170  Height = 35  AutoSize = True  Picture.Data = {	Position = poScreenCenter
PixelsPerInch = 96  TextHeight = 13  <> object Copyright: TLabel  Left = 82  Top = 81  Width = 207  Height = 13  Caption = 'Copyright 1983 - 2002  Borland International'  Font.Charset = ANSI_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  ParentFont = False end  <> object Bevel1: TBevel  Left = 86  Top = 158  Width = 251  Height = 2 end  <> object SKUName: TLabel  Left = 82  Top = 57  Width = 97  Height = 16  Caption = 'Using Delphi 7.0'  Font.Charset = ANSI_CHARSET  Font.Color = clBlack  Font.Height = -13  Font.Name = 'MS Sans Serif'  Font.Style = []  ParentFont = False end  <> object Image2: TImage  Left = 76  Top = 13  Width = 170  Height = 35  AutoSize = True  Picture.Data = {	
TextHeight = 13  <> object Copyright: TLabel Left = 82 Top = 81 Width = 207 Height = 13 Caption = 'Copyright 1983 - 2002 Borland International' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Bevel1: TBevel Left = 86 Top = 158 Width = 251 Height = 2 end <> object SKUName: TLabel Left = 82 Top = 57 Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	
<pre>&lt;&gt; object Copyright: TLabel Left = 82 Top = 81 Width = 207 Height = 13 Caption = 'Copyright 1983 - 2002 Borland International' Font.Charset = ANSI CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end &lt;&gt; object Bevel1: TBevel Left = 86 Top = 158 Width = 251 Height = 2 end &lt;&gt; object SKUName: TLabel Left = 82 Top = 57 Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end &lt;&gt; object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {</pre>	
Left = 82 Top = 81 Width = 207 Height = 13 Caption = 'Copyright 1983 - 2002 Borland International' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Bevel1: TBevel Left = 86 Top = 158 Width = 251 Height = 2 end <> object SKUName: TLabel Left = 82 Top = 57 Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	ĕ
Top = 81  Width = 207  Height = 13  Caption = 'Copyright 1983 - 2002  Borland International'  Font.Charset = ANSI_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  ParentFont = False end  <> object Bevel1: TBevel  Left = 86  Top = 158  Width = 251  Height = 2 end  <> object SKUName: TLabel  Left = 82  Top = 57  Width = 97  Height = 16  Caption = 'Using Delphi 7.0'  Font.Charset = ANSI_CHARSET  Font.Color = clBlack  Font.Height = -13  Font.Name = 'MS Sans Serif'  Font.Style = []  ParentFont = False end  <> object Image2: TImage  Left = 76  Top = 13  Width = 170  Height = 35  AutoSize = True  Picture.Data = {	
Width = 207  Height = 13  Caption = 'Copyright 1983 - 2002  Borland International'  Font.Charset = ANSI_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  ParentFont = False end  <> object Bevel1: TBevel  Left = 86  Top = 158  Width = 251  Height = 2 end  <> object SKUName: TLabel  Left = 82  Top = 57  Width = 97  Height = 16  Caption = 'Using Delphi 7.0'  Font.Charset = ANSI_CHARSET  Font.Color = clBlack  Font.Height = -13  Font.Name = 'MS Sans Serif'  Font.Style = []  ParentFont = False end  <> object Image2: TImage  Left = 76  Top = 13  Width = 170  Height = 35  AutoSize = True  Picture.Data = {	
Height = 13 Caption = 'Copyright 1983 - 2002 Borland International' Font.Charset = ANSI CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Bevel1: TBevel Left = 86 Top = 158 Width = 251 Height = 2 end <> object SKUName: TLabel Left = 82 Top = 57 Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	•
Caption = 'Copyright 1983 - 2002 Borland International'  Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Bevel1: TBevel Left = 86 Top = 158 Width = 251 Height = 2 end <> object SKUName: TLabel Left = 82 Top = 57 Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	
Borland International' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Bevel1: TBevel Left = 86 Top = 158 Width = 251 Height = 2 end <> object SKUName: TLabel Left = 82 Top = 57 Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Bevel1: TBevel Left = 86 Top = 158 Width = 251 Height = 2 end <> object SKUName: TLabel Left = 82 Top = 57 Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	
Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Bevel1: TBevel Left = 86 Top = 158 Width = 251 Height = 2 end <> object SKUName: TLabel Left = 82 Top = 57 Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	
Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Bevel1: TBevel Left = 86 Top = 158 Width = 251 Height = 2 end <> object SKUName: TLabel Left = 82 Top = 57 Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	
Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Bevel1: TBevel Left = 86 Top = 158 Width = 251 Height = 2 end <> object SKUName: TLabel Left = 82 Top = 57 Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	
Font.Style = [] ParentFont = False end <> object Bevel1: TBevel Left = 86 Top = 158 Width = 251 Height = 2 end <> object SKUName: TLabel Left = 82 Top = 57 Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	
ParentFont = False end  <> object Bevel1: TBevel Left = 86 Top = 158 Width = 251 Height = 2 end <> object SKUName: TLabel Left = 82 Top = 57 Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	
end  <> object Bevel1: TBevel  Left = 86  Top = 158  Width = 251  Height = 2  end  <> object SKUName: TLabel  Left = 82  Top = 57  Width = 97  Height = 16  Caption = 'Using Delphi 7.0'  Font.Charset = ANSI_CHARSET  Font.Color = clBlack  Font.Height = -13  Font.Name = 'MS Sans Serif'  Font.Style = []  ParentFont = False  end  <> object Image2: TImage  Left = 76  Top = 13  Width = 170  Height = 35  AutoSize = True  Picture.Data = {	
<pre>&lt;&gt; object Bevel1: TBevel Left = 86 Top = 158 Width = 251 Height = 2 end &lt;&gt; object SKUName: TLabel Left = 82 Top = 57 Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end &lt;&gt; object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {</pre>	ParentFont = False
Left = 86 Top = 158 Width = 251 Height = 2 end <> object SKUName: TLabel Left = 82 Top = 57 Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	
Top = 158  Width = 251  Height = 2 end  <> object SKUName: TLabel  Left = 82  Top = 57  Width = 97  Height = 16  Caption = 'Using Delphi 7.0'  Font.Charset = ANSI_CHARSET  Font.Color = clBlack  Font.Height = -13  Font.Name = 'MS Sans Serif'  Font.Style = []  ParentFont = False end  <> object Image2: TImage  Left = 76  Top = 13  Width = 170  Height = 35  AutoSize = True  Picture.Data = {	<> object Bevel1: TBevel
Width = 251 Height = 2 end  <> object SKUName: TLabel Left = 82 Top = 57 Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	Left = 86
Height = 2 end  <> object SKUName: TLabel  Left = 82  Top = 57  Width = 97  Height = 16  Caption = 'Using Delphi 7.0'  Font.Charset = ANSI_CHARSET  Font.Color = clBlack  Font.Height = -13  Font.Name = 'MS Sans Serif'  Font.Style = []  ParentFont = False end  <> object Image2: TImage  Left = 76  Top = 13  Width = 170  Height = 35  AutoSize = True  Picture.Data = {	Top = 158
end  <> object SKUName: TLabel  Left = 82  Top = 57  Width = 97  Height = 16  Caption = 'Using Delphi 7.0'  Font.Charset = ANSI_CHARSET  Font.Color = clBlack  Font.Height = -13  Font.Name = 'MS Sans Serif'  Font.Style = []  ParentFont = False  end  <> object Image2: TImage  Left = 76  Top = 13  Width = 170  Height = 35  AutoSize = True  Picture.Data = {	Width = 251
<pre>&lt;&gt; object SKUName: TLabel   Left = 82   Top = 57   Width = 97   Height = 16   Caption = 'Using Delphi 7.0'   Font.Charset = ANSI_CHARSET   Font.Color = clBlack   Font.Height = -13   Font.Name = 'MS Sans Serif'   Font.Style = []   ParentFont = False   end   &lt;&gt; object Image2: TImage   Left = 76   Top = 13   Width = 170   Height = 35   AutoSize = True   Picture.Data = {</pre>	Height = 2
Left = 82  Top = 57  Width = 97  Height = 16  Caption = 'Using Delphi 7.0'  Font.Charset = ANSI_CHARSET  Font.Color = clBlack  Font.Height = -13  Font.Name = 'MS Sans Serif'  Font.Style = []  ParentFont = False end  <> object Image2: TImage  Left = 76  Top = 13  Width = 170  Height = 35  AutoSize = True  Picture.Data = {	end
Left = 82  Top = 57  Width = 97  Height = 16  Caption = 'Using Delphi 7.0'  Font.Charset = ANSI_CHARSET  Font.Color = clBlack  Font.Height = -13  Font.Name = 'MS Sans Serif'  Font.Style = []  ParentFont = False end  <> object Image2: TImage  Left = 76  Top = 13  Width = 170  Height = 35  AutoSize = True  Picture.Data = {	<> object SKUName: TLabel
Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	·
Width = 97 Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	
Height = 16 Caption = 'Using Delphi 7.0' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	
Caption = 'Using Delphi 7.0' Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	
Font.Charset = ANSI_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	
Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	
Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	<del>-</del>
Font.Name = 'MS Sans Serif' Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	
Font.Style = [] ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	
ParentFont = False end <> object Image2: TImage Left = 76 Top = 13 Width = 170 Height = 35 AutoSize = True Picture.Data = {	
end <> object Image2: TImage  Left = 76  Top = 13  Width = 170  Height = 35  AutoSize = True  Picture.Data = {	
<> object Image2: TImage  Left = 76  Top = 13  Width = 170  Height = 35  AutoSize = True  Picture.Data = {	
Left = 76	
Top = 13  Width = 170  Height = 35  AutoSize = True  Picture.Data = {	
Width = 170 Height = 35 AutoSize = True Picture.Data = {	
Height = 35 AutoSize = True Picture.Data = {	
AutoSize = True Picture.Data = {	Width = 170
Picture.Data = {	Č
	AutoSize = True
	Picture.Data = {
	000}

Classes, Graphics, Controls, Forms,
Dialogs,
StdCtrls, jpeg, ExtCtrls, FileCtrl,
ComCtrls, Menus, printers;
type
TForm1 = class(TForm)
Panel1: TPanel;
DirectoryListBox1:
TDirectoryListBox;
FileListBox1: TFileListBox;
Panel3: TPanel;
DriveComboBox1:
TDriveComboBox;
Scale: TComboBox;
PixelFormat: TComboBox;
ColorSpace: TComboBox;
Performance: TComboBox;
ProgressiveDisplay: TCheckBox;
IncrementalDisplay: TCheckBox;
MainMenu1: TMainMenu;
File1: TMenuItem;
Open1: TMenuItem;
N1: TMenuItem;
Print1: TMenuItem;
PrinterSetup1: TMenuItem;
N2: TMenuItem;
Exit1: TMenuItem;
OpenDialog1: TOpenDialog;
PrinterSetupDialog1:
1 0
TPrinterSetupDialog;
PrintDialog1: TPrintDialog;
ScrollBox1: TScrollBox;
Image1: TImage;
<>procedure
FileListBox1DblClick(Sender:
TObject);
<>procedure
SetJPEGOptions(Sender: TObject);
<>procedure
FormCreate(Sender: TObject);
<>procedure
ProgressUpdate(Sender: TObject;
Stage: TProgressStage;
PercentDone: Byte; RedrawNow:
Boolean; const R: TRect; const Msg:
string);
<>procedure
Open1Click(Sender: TObject);

# القسم الرابع المستكشف في دلفي

Procedure\$20
الهلف التنفيذي لبرنامج
(Jpeg)
إعداد
علاء الدين اللباد
واتف ۷۳۷۱۹۹۱
program jpegproj;
wsos
Forms,
jpegdemo in 'jpegdemo.pas'
{Form1};
Arounts,
{\$R *.RES}
begin
Application.Initialize;
Application.CreateForm(TForm1,
Form1);
Application.Run;
end.
111111111111111111111111111111111111111
unit jpegdemo;
interface
uses
Windows, Messages, SysUtils,

var	<>procedure
Temp: Boolean;	Print1Click(Sender: TObject);
begin	<>procedure
Temp := Image1.Picture.Graphic is	PrinterSetup1Click(Sender:
TJPEGImage;	TObject);
if Temp then	<>procedure
with	Exit1Click(Sender: TObject);
TJPEGImage(Image1.Picture.Grap	<>procedure
hic) do	ScrollBox1Click(Sender: TObject);
begin	private
PixelFormat :=	{ Private declarations }
TJPEGPixelFormat(Self.PixelForma	public
t.ItemIndex);	{ Public declarations }
Scale :=	<>procedure OpenFile(const
TJPEGScale(Self.Scale.ItemIndex);	Filename: string);
Grayscale :=	end;
Boolean(Colorspace.ItemIndex);	
Performance :=	var
TJPEGPerformance(Self.Performan	Form1: TForm1;
ce.ItemIndex);	,
ProgressiveDisplay :=	implementation
Self.ProgressiveDisplay.Checked;	
end;	{\$R *.DFM}
Scale.Enabled := Temp;	(the little state of the l
PixelFormat.Enabled := Temp;	<>procedure
Colorspace.Enabled := Temp;	TForm1.OpenFile(const Filename:
Performance.Enabled := Temp;	string);
ProgressiveDisplay.Enabled :=	begin
Temp	try
and	ti y
TJPEGImage(Image1.Picture.Grap	Image1.Picture.LoadFromFile(Filen
hic).ProgressiveEncoding;	ame);
Image1.IncrementalDisplay :=	except
IncrementalDisplay.Checked;	on EInvalidGraphic do
end;	Image1.Picture.Graphic := nil;
· · · · · · · · · · · · · · · · · · ·	1
<>procedure	end;
TForm1.FormCreate(Sender:	SetJPEGOptions(self);
TObject);	end;
begin	
Scale.ItemIndex := 0;	<>procedure
PixelFormat.ItemIndex := 0;	TForm1.FileListBox1DblClick(Send
	er: TObject);
Colorspace.ItemIndex := 0;	begin
Performance.ItemIndex := 0;	OpenFile(FileListbox1.Filename);
OpenDialog1.Filter :=	end;
GraphicFilter(TGraphic);	
FileListbox1.Mask :=	<>procedure
GraphicFileMask(TGraphic);	TForm1.SetJPEGOptions(Sender:
Image1.OnProgress :=	TObject);

begin	F
if OutputWidth < OutputHeight	e
then	
begin	<
OutputHeight :=	Т
Printer.PageHeight;	Т
OutputWidth := OutputHeight	
* AspectRatio;	E
end	S
else	b
begin	
OutputWidth :=	
Printer.PageWidth;	I
OutputHeight := OutputWidth /	
AspectRatio;	
end	e
end;	
if OutputWidth >	
Printer.PageWidth then	<
begin	
OutputWidth :=	
Printer.PageWidth;	ŀ
OutputHeight := OutputWidth /	<u> </u>
AspectRatio;	
end;	
if OutputHeight >	e
Printer.PageHeight then	
begin	<
OutputHeight :=	
Printer.PageHeight;	
OutputWidth := OutputHeight *	1
AspectRatio;	H'
end;	
enu,	5
Printer.Canvas.StretchDraw(Rect(0,	-
0,	ŀ
Trunc(OutputWidth),	Ι,
` = '.'	I
Trunc(OutputHeight)),	
Image1.Picture.Graphic);	
finally	_
Printer.EndDoc;	I
end;	
end;	I
<>procedure	(
TForm1.PrinterSetup1Click(Sender:	
TObject);	I
begin	
PrinterSetupDialog1.Execute;	F

ProgressUpdate;
end;
<>procedure
TForm1.ProgressUpdate(Sender:
TObject; Stage: TProgressStage;
PercentDone: Byte; RedrawNow:
Boolean; const R: TRect; const Msg:
string);
begin
if Stage = psRunning then
Caption :=
Format('%d%%',[PercentDone])
else
Caption := 'Form1';
end;
<>procedure
TForm1.Open1Click(Sender:
TObject);
begin
if OpenDialog1.Execute then
ii OpenDialog1.Execute then
On an Eile (On an Diale of Eile Name)
OpenFile(OpenDialog1.FileName);
end;
_
<>procedure
TForm1.Print1Click(Sender:
TObject);
var
AspectRatio: Single;
OutputWidth, OutputHeight:
Single;
begin
if not PrintDialog1.Execute then
Exit;
Printer.BeginDoc;
try
OutputWidth :=
Image1.Picture.Width;
OutputHeight :=
Image1.Picture.Height;
AspectRatio := OutputWidth /
OutputHeight;
if (OutputWidth <
Printer.PageWidth) and
(OutputHeight <
Printer.PageHeight) then
i i i i i i i i i i i i i i i i i i i

<b>Top</b> = 27
Width = 124
Height = 349
Align = alLeft
TabOrder = 0
<>Object DirectoryListBox1:
TDirectoryListBox
Left = 1
Top = 1
Width = 122
Height = 79
Align = alTop
Ctl3D = True
FileList = FileListBox1
ItemHeight = 16
ParentCtl3D = False
TabOrder = 0
end
<>Object FileListBox1:
TFileListBox
Left = 1
Top = 80
Width = 122
Height = 268
Align = alClient
ItemHeight = 13
TabOrder = 1
OnClick = FileListBox1DblClick
end
end
<>Object Panel3: TPanel
Left = 0
Top = 0
Width = 488
Height = 27
Align = alTop
TabOrder = 1
<pre>&lt;&gt;Object DriveComboBox1:</pre>
TDriveComboBox
Left = 7
Top = 3
Width = 117
Height = 19
DirList = DirectoryListBox1
TabOrder = 0
end
<>Object Scale: TComboBox
Left = 127

end;
<>procedure
TForm1.Exit1Click(Sender:
TObject);
begin
Close;
end;
<>procedure
TForm1.ScrollBox1Click(Sender:
TObject);
begin
end;

<>Objects26
الملف النصي لبرنامج
(Jpeg)
عامدا
علاء الدين اللباد
الله ۱۷۹۵۸۵۶۹۰
<>Object Form1: TForm1
Left = 225
Top = 115
AutoScroll = False
Caption = 'Form1'
ClientHeight = 376
ClientWidth = 488
Color = clBtnFace
Font.Charset = ANSI_CHARSET
Font.Color = clWindowText
Font.Height = -10
Font.Name = 'MS Sans Serif'
Font.Style = []
Menu = MainMenu1
OldCreateOrder = True
Position = poDefaultPosOnly
ShowHint = True
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<>Object Panel1: TPanel
Left = 0

TComboBox
Left = 283
Top = 3
Width = 75
Height = 21
Hint = 'Performance'
Style = csDropDownList
Enabled = False
ItemHeight = 13 TabOrder = 4
OnClick = SetJPEGOptions
Items.Strings = (
'Quality'
'Speed')
end
<>Object ProgressiveDisplay:
TCheckBox
Left = 364
Top = 1
Width = 124
Height = 14
Caption = 'Progressive JPEG'
Enabled = False
Font.Charset =
ANSI_CHARSET
Font.Color = clWindowText
Font.Height = -9
Font.Name = 'arial'
Font.Style = []
ParentFont = False
TabOrder = 5
OnClick = SetJPEGOptions
end
<>Object
IncrementalDisplay: TCheckBox
Left = 364
Top = 12
Width = 137
Height = 14
Caption = 'Incremental Display'
Font.Charset =
ANSI CHARSET
Font.Color = clWindowText
Font.Height = -9
Font.Name = 'Arial'
Font.Style = []
ParentFont = False
TabOrder = 6

Top = 3
Width = 36
Height = 21
Hint = 'Scale'
Style = csDropDownList
Enabled = False
ItemHeight = 13
TabOrder = 1
OnClick = SetJPEGOptions
Items.Strings = (
'1:1'
'1:2'
'1:4'
'1:8')
end
<>Object PixelFormat: TComboBox
Left = 166
Top = 3
Width = 49
Height = 21
Hint = 'Pixel Format'
Style = csDropDownList
Enabled = False
ItemHeight = 13
TabOrder = 2
OnClick = SetJPEGOptions
Items.Strings = (
'24 bit'
'8 bit')
end
<>Object ColorSpace:
TComboBox
Left = 218
Top = 3
Width = 62
Height = 21
Hint = 'Color space'
Style = csDropDownList
Enabled = False
ItemHeight = 13
TabOrder = 3
OnClick = SetJPEGOptions
_
Items.Strings = ( 'RGB'
'Grayscale')
end
<>Object Performance:

Ch and Cn4 = 2205(
ShortCut = 32856
OnClick = Exit1Click
end
end
end
<>Object OpenDialog1:
TOpenDialog
Options = [ofPathMustExist,
ofFileMustExist]
Left = 160
Top = 225
end
<>Object
PrinterSetupDialog1:
TPrinterSetupDialog
Left = 168
Top = 305
end
<>Object PrintDialog1:
TPrintDialog
Left = 96
Top = 377
end
end

end
end
<>Object ScrollBox1:
TScrollBox
Left = 124
Top = 27
Width = 364
Height = 349
Align = alClient
TabOrder = 2
<>Object Image1: TImage
Left = 0
Top = 0
Width = 85
Height = 85
AutoSize = True
end
end
<>Object MainMenu1:
TMainMenu
Left = 152
Top = 145
<>Object File1: TMenuItem
Caption = '&File'
<>Object Open1:
TMenuItem
Caption = '&Open'
ShortCut = 114
OnClick = Open1Click
end
<>Object N1: TMenuItem
Caption = '-'
end
<>Object Print1:
TMenuItem
Caption = '&Print'
OnClick = Print1Click
end
<>Object PrinterSetup1:
TMenuItem
Caption = 'Printer & Setup'
OnClick = PrinterSetup1Click
end
<>Object N2: TMenuItem
Caption = '-'
end
<>Object Exit1: TMenuItem
· ·
Caption = 'E&xit'

A44-91-4 T-4
Attributes: Integer;
ModDate: string;
end;
TForm1 - alogg(TForm)
TForm1 = class(TForm)
ListView: TListView;
CoolBar1: TCoolBar;
ToolBar2: TToolBar;
ToolbarImages: TImageList;
btnBrowse: TToolButton;
btnLargeIcons: TToolButton;
btnSmallIcons: TToolButton;
btnList: TToolButton;
btnReport: TToolButton;
cbPath: TComboBox;
ToolButton3: TToolButton;
PopupMenu1: TPopupMenu;
btnBack: TToolButton;
<> procedure
FormCreate(Sender: TObject);
<> procedure
ListViewData(Sender: TObject;
Item: TListItem);
<> procedure
btnBrowseClick(Sender: TObject);
<> procedure
cbPathKeyDown(Sender: TObject;
var Key: Word;
Shift: TShiftState);
<> procedure
cbPathClick(Sender: TObject);
<> procedure
btnLargeIconsClick(Sender:
TObject);
<> procedure
ListViewDblClick(Sender: TObject);
<> procedure
ListViewDataHint(Sender: TObject;
StartIndex,
EndIndex: Integer);
<> procedure
ListViewKeyDown(Sender:
TObject; var Key: Word;
Shift: TShiftState);
<pre>&lt;&gt; procedure ListViewDataFind(Sandary TObject)</pre>
ListViewDataFind(Sender: TObject;
Find: TItemFind;
const FindString: String; const
FindPosition: TPoint;

. 1 . 1 . 2 11 21 11
الملف التنفيذي لبرنامج
Virtual Listview
Procedure40
)
alae į
علاء الدين اللباد
8اتة. ۷۳٬۵۷۵۶۶۹۰
unit VListView;
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs, ComCtrls, StdCtrls, ToolWin,
ShlObj, ImgList, Menus;
Smooj, inglist, wenus,
type
PShellItem = ^TShellItem;
TShellItem = record
FullID,
ID: PItemIDList;
Empty: Boolean;
DisplayName,
TypeName: string;
ImageIndex,
Size,

FindData: Pointer;	
uses ShellAPI, ActiveX, ComObj, Integer; Direction:	~ ••••
CommCtrl, FileCtrl;  TSearchDirection;	
Wrap: Boolean; van	r Index:
//PIDL MANIPULATION Integer);	
<> procedure	
<> procedure DisposePIDL(ID: ListViewCustomDrawl	Item(Sender:
PItemIDList); TCustomListView;	(
var Item: TListItem; St	tate:
Malloc: IMalloc; TCustomDrawState; va	
begin DefaultDraw: Boolean)	);
if ID = nil then Exit; <> procedure	
OLECheck(SHGetMalloc(Malloc)); ListViewCustomDrawS	SubItem(Send
Malloc.Free(ID); er: TCustomListView;	•
end; Item: TListItem; Su	ubItem:
Integer; State: TCuston	mDrawState;
function CopyITEMID(Malloc: var DefaultDraw: B	
IMalloc; ID: PItemIDList): <> procedure	
PItemIDList; btnBackClick(Sender:	TObject);
begin <> procedure	
Result := Malloc, Alloc(ID^, mkid, cb   Form1Close(Sender: T	
+ SizeOf(ID^.mkid.cb));  Action: TCloseAction);	,
CopyMemory(Result, ID, private	
ID^.mkid.cb + FPIDL: PItemIDList	t;
SizeOf(ID^.mkid.cb)); FIDList: TList;	
end; FIShellFolder,	
FIDesktopFolder: IS	ShellFolder;
function NextPIDL(IDList: FPath: string;	
PItemIDList): PItemIDList; <> procedure S	SetPath(const
begin Value: string); overload	
Result := IDList; <> procedure S	SetPath(ID:
Inc(PChar(Result), PItemIDList); overload	d;
IDList^.mkid.cb); <> procedure	
end; PopulateIDList(ShellFo	older:
IShellFolder);	
function GetPIDLSize(IDList: <> procedure C	ClearIDList;
PItemIDList): Integer; <> procedure	
begin CheckShellItems(Start	Index,
Result := 0; EndIndex: Integer);	
if Assigned(IDList) then function ShellItem(I	Index:
begin Integer): PShellItem;	
Result := end;	
SizeOf(IDList^.mkid.cb);	
while IDList^.mkid.cb $\Leftrightarrow$ 0 do var	
begin Form1: TForm1;	
Result := Result +	
IDList^.mkid.cb; implementation	
IDList := NextPIDL(IDList);	
end; {\$R *.dfm}	

if Assigned(Result) then	
CopyMemory(Result, IDList,	
Size);	
end;	
function ConcatPIDLs(IDList1,	
IDList2: PItemIDList):	_
PItemIDList;	_
var	
cb1, cb2: Integer;	
begin	
if Assigned(IDList1) then	
cb1 := GetPIDLSize(IDList1) -	_
SizeOf(IDList1\cdots.mkid.cb)	F
else	
cb1 := 0;	-
	-
cb2 := GetPIDLSize(IDList2);	-
cuz :- Getr IDLSize(IDListz);	F
Desult - Cuesta DIDI (ab.1 + ab.2).	F
Result := CreatePIDL(cb1 + cb2);	-
if Assigned(Result) then	-
begin	
if Assigned(IDList1) then	
CopyMemory(Result, IDList1,	
cb1);	L
CopyMemory(PChar(Result) +	L
cb1, IDList2, cb2);	
end;	
end;	
//SHELL FOLDER ITEM INFO	
6 4	
function	<u> </u>
GetDisplayName(ShellFolder:	
IShellFolder; PIDL: PItemIDList;	
ForParsing: Boolean):	
string;	
var	
StrRet: TStrRet;	
P: PChar;	Ī
Flags: Integer;	
begin	Ī
Result := '';	
if ForParsing then	F
Flags := SHGDN_FORPARSING	
else	F
Flags := SHGDN NORMAL;	-
	-

_
end;
end;
<> procedure
StripLastID(IDList: PItemIDList);
var
MarkerID: PItemIDList;
begin
MarkerID := IDList;
if Assigned(IDList) then
begin
while IDList.mkid.cb
begin
MarkerID := IDList;
IDList := NextPIDL(IDList);
end;
MarkerID.mkid.cb := 0;
end;
end;
function CreatePIDL(Size: Integer):
PItemIDList;
var
Malloc: IMalloc;
HR: HResult;
begin
9
Result := nil;
TID SHC (M.H. (M.H.)
HR := SHGetMalloc(Malloc);
if Failed(HR) then
Exit;
try
<b>Result</b> := Malloc.Alloc(Size);
if Assigned(Result) then
FillChar(Result^, Size, 0);
finally
end;
end;
, , , , , , , , , , , , , , , , , , ,
function CopyPIDL(IDList:
PItemIDList): PItemIDList;
, ,
Sign. Integrate
Size: Integer;
begin
Size := GetPIDLSize(IDList);
Result := CreatePIDL(Size);

var	
Flags: UINT;	Sh
begin	L,
Flags := SFGAO_FOLDER;	Ca
ShellFolder.GetAttributesOf(1, ID,	- 5
Flags);	100
Result := SFGAO_FOLDER and	lSt
$Flags \diamondsuit 0;$	
end;	
	a
function ListSortFunc(Item1, Item2:	Siz
Pointer): Integer;	
begin	PI
Result :=	
SmallInt(Form1.FIShellFolder.Com	,
pareIDs(	
0,	eı
,	en
PShellItem(Item1).ID,	en
PShellItem(Item2).ID	C
));	fu
end;	PI
	Bo
{TForm1}	va
	F
//GENERAL FORM METHODS	F
	be
<> procedure	
<> procedure TForm1.FormCreate(Sender:	F
•	#0
TForm1.FormCreate(Sender:	#0 F
TForm1.FormCreate(Sender: TObject); var	#0 F SH
TForm1.FormCreate(Sender: TObject); var FileInfo: TSHFileInfo;	F #0 F SH SH
TForm1.FormCreate(Sender: TObject); var FileInfo: TSHFileInfo; ImageListHandle: THandle;	F #0 F SH SH it
TForm1.FormCreate(Sender: TObject); var FileInfo: TSHFileInfo; ImageListHandle: THandle; NewPIDL: PItemIDList;	F #0 F SH SH
TForm1.FormCreate(Sender: TObject); var FileInfo: TSHFileInfo; ImageListHandle: THandle;	F #0 F SH SH if
TForm1.FormCreate(Sender: TObject); var FileInfo: TSHFileInfo; ImageListHandle: THandle; NewPIDL: PItemIDList; begin	F #0 F SH SH if SH
TForm1.FormCreate(Sender: TObject); var FileInfo: TSHFileInfo; ImageListHandle: THandle; NewPIDL: PItemIDList; begin OLECheck(SHGetDesktopFolder(FI	F #0 F SH SH if SH SH e
TForm1.FormCreate(Sender: TObject); var FileInfo: TSHFileInfo; ImageListHandle: THandle; NewPIDL: PItemIDList; begin OLECheck(SHGetDesktopFolder(FIDesktopFolder));	F #0 F SH SH if SH e SH
TForm1.FormCreate(Sender: TObject); var FileInfo: TSHFileInfo; ImageListHandle: THandle; NewPIDL: PItemIDList; begin  OLECheck(SHGetDesktopFolder(FIDesktopFolder)); FIShellFolder := FIDesktopFolder;	F #0 F SH SH if SH if SH SH SH
TForm1.FormCreate(Sender: TObject); var FileInfo: TSHFileInfo; ImageListHandle: THandle; NewPIDL: PItemIDList; begin  OLECheck(SHGetDesktopFolder(FIDesktopFolder)); FIShellFolder := FIDesktopFolder; FIDList := TList.Create;	F #0 F SH SH if SH if SH SH SH
TForm1.FormCreate(Sender: TObject); var FileInfo: TSHFileInfo; ImageListHandle: THandle; NewPIDL: PItemIDList; begin  OLECheck(SHGetDesktopFolder(FIDesktopFolder)); FIShellFolder := FIDesktopFolder; FIDList := TList.Create; ImageListHandle :=	F #0 F SH SH if SH e SH
TForm1.FormCreate(Sender: TObject); var FileInfo: TSHFileInfo; ImageListHandle: THandle; NewPIDL: PItemIDList; begin  OLECheck(SHGetDesktopFolder(FIDesktopFolder)); FIShellFolder:=FIDesktopFolder; FIDList:=TList.Create; ImageListHandle:= SHGetFileInfo('C:\',	F #0 F SH SH if SH e SH
TForm1.FormCreate(Sender: TObject); var FileInfo: TSHFileInfo; ImageListHandle: THandle; NewPIDL: PItemIDList; begin  OLECheck(SHGetDesktopFolder(FIDesktopFolder)); FIShellFolder := FIDesktopFolder; FIDList := TList.Create; ImageListHandle := SHGetFileInfo('C:\', 0,	F #0 F SH SH if SH SSF SSF SSF SSF SSF SSF SSF SSF
TForm1.FormCreate(Sender: TObject); var FileInfo: TSHFileInfo; ImageListHandle: THandle; NewPIDL: PItemIDList; begin  OLECheck(SHGetDesktopFolder(FIDesktopFolder)); FIShellFolder := FIDesktopFolder; FIDList := TList.Create; ImageListHandle := SHGetFileInfo('C:\',  0, FileInfo,	F #0 F SH SH iff SH SH SH E SH R
TForm1.FormCreate(Sender: TObject); var  FileInfo: TSHFileInfo; ImageListHandle: THandle; NewPIDL: PItemIDList; begin  OLECheck(SHGetDesktopFolder(FIDesktopFolder)); FIShellFolder := FIDesktopFolder; FIDList := TList.Create; ImageListHandle := SHGetFileInfo('C:\', 0,	F #0 F SH SH if SH SH SH SH R
TForm1.FormCreate(Sender: TObject); var FileInfo: TSHFileInfo; ImageListHandle: THandle; NewPIDL: PItemIDList; begin  OLECheck(SHGetDesktopFolder(FIDesktopFolder)); FIShellFolder := FIDesktopFolder; FIDList := TList.Create; ImageListHandle := SHGetFileInfo('C:\',  0, FileInfo,	F #0 F SH SH iff SH el SH R
TForm1.FormCreate(Sender: TObject); var  FileInfo: TSHFileInfo; ImageListHandle: THandle; NewPIDL: PItemIDList; begin  OLECheck(SHGetDesktopFolder(FIDesktopFolder)); FIShellFolder:=FIDesktopFolder; FIDList:=TList.Create; ImageListHandle:= SHGetFileInfo('C:\', 0, FileInfo, SizeOf(FileInfo),	F #00 F SH SH iff SH

ShellFolder.GetDisplayNameOf(PID
L, Flags, StrRet);
case StrRet.uType of
STRRET CSTR:
SetString(Result, StrRet.cStr,
lStrLen(StrRet.cStr));
STRRET OFFSET:
begin
P :=
@PIDL.mkid.abID[StrRet.uOffset -
SizeOf(PIDL.mkid.cb)];
SetString(Result, P,
PIDL.mkid.cb - StrRet.uOffset);
end;
STRRET WSTR:
Result := StrRet.pOleStr;
end;
end;
function GetShellImage(PIDL:
PItemIDList; Large, Open:
Boolean): Integer;
var
FileInfo: TSHFileInfo;
Flags: Integer;
begin
FillChar(FileInfo, SizeOf(FileInfo),
#0);
Flags := SHGFI_PIDL or
SHGFI_SYSICONINDEX or
SHGFI_ICON;
if Open then Flags := Flags or
SHGFI_OPENICON;
if Large then Flags := Flags or
SHGFI_LARGEICON
else Flags := Flags or
SHGFI_SMALLICON;
SHGetFileInfo(PChar(PIDL),
0,
FileInfo,
SizeOf(FileInfo),
Flags);
Result := FileInfo.iIcon;
end;
function IsFolder(ShellFolder:
IShellFolder; ID: PItemIDList):
Boolean;

cbPath.Text[Length(cbPath.Text)] =	LVM_SETIMAGELIST,
':' then	LVSIL_SMALL, ImageListHandle);
cbPath.Text := cbPath.Text + '\';	
SetPath(cbPath.Text);	ImageListHandle :=
Key := 0;	SHGetFileInfo('C:\',
end;	0,
end;	FileInfo,
	SizeOf(FileInfo),
<> procedure	
TForm1.cbPathClick(Sender:	SHGFI SYSICONINDEX or
TObject);	SHGFI_LARGEICON);
var	-
I: Integer;	SendMessage(ListView.Handle,
begin	LVM SETIMAGELIST,
I :=	LVSIL NORMAL,
cbPath.Items.IndexOf(cbPath.Text);	ImageListHandle);
if $I \ge 0$ then	OLECheck(
	SHGetSpecialFolderLocation(
SetPath(PItemIDList(cbPath.Items.	Application.Handle,
Objects[I]))	CSIDL DRIVES,
else	NewPIDL)
SetPath(cbPath.Text);	);
end;	SetPath(NewPIDL);
chu,	ActiveControl := cbPath;
<> procedure	cbPath.SelStart := 0;
TForm1.btnLargeIconsClick(Sender	cbPath.SelLength :=
: TObject);	Length(cbPath.Text);
begin	end;
ListView.ViewStyle :=	cnu,
TViewStyle((Sender as	<> procedure
TComponent).Tag);	TForm1.btnBrowseClick(Sender:
end;	TObject);
cnu,	<b>"</b>
<> procedure	S: string;
TForm1.ListViewDblClick(Sender:	
TObject);	begin S := ";
• / /	,
RootPIDL,	if SelectDirectory('Select
ID: PItemIDList;	Directory', ", S) then
,	SetPath(S);
begin if ListView.Selected ⇔ nil then	end;
	/
begin	<> procedure TEarm 1 ab Dath V ov Down (Sandam)
ID := Shall tarr (Light Vierry Salacted Index) I	TForm1.cbPathKeyDown(Sender:
ShellItem(ListView.Selected.Index).I	TObject; var Key: Word;
D;	Shift: TShiftState);
if not IsFolder(FIShellFolder, ID)	begin
then Exit;	if Key = VK_RETURN then
RootPIDL :=	begin
ConcatPIDLs(FPIDL, ID);	if

EnumList: IEnumIDList;	SetPath(RootPIDL);
NumIDs: LongWord;	end;
SaveCursor: TCursor;	end;
ShellItem: PShellItem;	
begin	function TForm1.ShellItem(Index:
SaveCursor := Screen.Cursor;	Integer): PShellItem;
try	begin
Screen.Cursor := crHourglass;	Result :=
OleCheck(	PShellItem(FIDList[Index]);
ShellFolder.EnumObjects(	end;
Application.Handle,	
Flags,	<> procedure
EnumList)	TForm1.ListViewKeyDown(Sender:
);	TObject; var Key: Word;
	Shift: TShiftState);
FIShellFolder := ShellFolder;	begin
ClearIDList;	case Key of
while EnumList.Next(1, ID,	VK RETURN:
NumIDs) = S OK do	ListViewDblClick(Sender);
begin	VK BACK:
ShellItem := New(PShellItem);	btnBackClick(Sender);
ShellItem.ID := ID;	end;
ShellItem.DisplayName :=	end;
GetDisplayName(FIShellFolder, ID,	Citu,
False);	//SHELL-RELATED ROUTINES.
ShellItem.Empty := True;	#SHEEL-RELATED ROUTHVES.
FIDList.Add(ShellItem);	<> procedure
end;	TForm1.ClearIDList;
Chu,	var
FIDList.Sort(ListSortFunc);	I: Integer;
riblist.sort(Listsortruite),	begin
//We need to tell the ListView how	for I := 0 to FIDList.Count-1 do
many items it has.	begin
ListView.Items.Count :=	DisposePIDL(ShellItem(I).ID);
FIDList.Count;	Dispose(ShellItem(I));
riblist.Count,	end;
ListView.Repaint;	FIDList.Clear;
• '	·
finally Screen.Cursor := SaveCursor;	end;
	z nuo anduma
end;	<> procedure TForm1.PopulateIDList(ShellFolder
end;	
/ > munandarus	: IShellFolder);
<> procedure	const
TForm1.SetPath(const Value:	Flags = SHCONTF_FOLDERS or SHCONTF_NONFOLDERS or
string);	SHCONTF_NONFOLDERS OF SHCONTF INCLUDEHIDDEN;
D. DWideChare	<del>-</del>
P: PWideChar; NewPIDL: PItemIDList;	var ID: PItemIDList;
rewrith: Pitemilliast:	լ ութ. ուսաութել,

cbPath.Text := cbPath.Items[0];	Flags,
end	NumChars: LongWord;
else begin	begin
cbPath.ItemIndex := Index;	NumChars := Length(Value);
cbPath.Text :=	Flags := 0;
cbl atil. Text .— cbPath.Items[cbPath.ItemIndex];	P := StringToOleStr(Value);
end;	1 String rootestr (value),
Ciru,	OLECheck(
if ListView.Items.Count > 0 then	OLECHECK
begin	FIDesktopFolder.ParseDisplayName
ListView.Selected :=	(
ListView.Items[0];	Application. Handle,
ListView.Selected.Focused :=	nil,
True;	P,
1140,	NumChars,
ListView.Selected.MakeVisible(False	NewPIDL,
);	Flags)
end;	);
finally	SetPath(NewPIDL);
ListView.Items.EndUpdate;	end;
end;	Citu,
end;	<> procedure
	TForm1.SetPath(ID: PItemIDList);
//ROUTINES FOR MANAGING	var
VIRTUAL DATA	Index: Integer;
	NewShellFolder: IShellFolder;
<> procedure	begin
TForm1.CheckShellItems(StartInde	OLECheck(
x, EndIndex: Integer);	FIDesktopFolder.BindToObject(
	ID,
function ValidFileTime(FileTime:	nil,
TFileTime): Boolean;	IID IShellFolder,
begin	Pointer(NewShellFolder))
Result :=	);
(FileTime.dwLowDateTime ← 0) or	
(FileTime.dwHighDateTime <> 0);	ListView.Items.BeginUpdate;
end;	try
	PopulateIDList(NewShellFolder);
var	FPIDL := ID;
FileData: TWin32FindData;	FPath :=
FileInfo: TSHFileInfo;	GetDisplayName(FIDesktopFolder,
SysTime: TSystemTime;	FPIDL, True);
I: Integer;	Index :=
LocalFileTime: TFILETIME;	cbPath.Items.IndexOf(FPath);
begin	if (Index < 0) then
//Here all the data that wasn't	begin
initialized in PopulateIDList is	cbPath.Items.InsertObject(0,
//filled in.	FPath, Pointer(FPIDL));
for I := StartIndex to EndIndex do	

me, SysTime) then	begin
try	if ShellItem(I)^.Empty then
ModDate :=	with ShellItem(I)^ do
DateTimeToStr(SystemTimeToDate	begin
Time(SysTime))	FullID := ConcatPIDLs(FPIDL,
except	ID);
on EConvertError do	ImageIndex :=
ModDate := ";	GetShellImage(FullID,
end	ListView.ViewStyle = vsIcon, False);
else	
ModDate := ";	//File Type
	SHGetFileInfo(
//Attributes	PChar(FullID),
Attributes :=	0,
FileData.dwFileAttributes;	FileInfo,
The butter with the first the state of the s	SizeOf(FileInfo),
//Flag this record as complete.	SHGFI TYPENAME or
Empty := False;	SHGFI PIDL
end;	).
end;	TypeName :=
	FileInfo.szTypeName;
end;	rneimo.sz i ypervame,
z S www.co.duwe	//Get File info from Windows
<> procedure	
TForm1.ListViewDataHint(Sender:	FillChar(FileData,
TObject; StartIndex,	SizeOf(FileData), #0);
EndIndex: Integer);	SHGetDataFromIDList(
begin	FIShellFolder,
//OnDataHint is called before	ID,
OnData. This gives you a chance to	SHGDFIL_FINDDATA,
//initialize only the data structures	@FileData,
that need to be drawn.	SizeOf(FileData)
//You should keep track of which	);
items have been initialized so no	
//extra work is done.	//File Size, in KB
if (StartIndex > FIDList.Count) or	Size := (FileData.nFileSizeLow +
(EndIndex > FIDList.Count) then	1023 ) div 1024;
Exit;	if Size = 0 then Size := 1;
CheckShellItems(StartIndex,	
EndIndex);	//Modified Date
end;	FillChar(LocalFileTime,
_	SizeOf(TFileTime), #0);
<> procedure	with FileData do
TForm1.ListViewData(Sender:	if
TObject; Item: TListItem);	ValidFileTime(ftLastWriteTime)
var	and
Attrs: string;	FileTimeToLocalFileTime(ftLastWr
begin	iteTime, LocalFileTime)
//OnData gets called once for each	and
item for which the ListView needs	FileTimeToSystemTime(LocalFileTi

FindPosition: TPoint; FindData: Pointer;	//data. If the ListView is in Report
StartIndex: Integer; Direction:	View, be sure to add the subitems.  //Item is a "dummy" item whose
TSearchDirection; Wrap: Boolean;	only valid data is it's index which
var Index: Integer);	//is used to index into the
3 //	
//OnDataFind gets called in response	underlying data. if (Item.Index > FIDList.Count)
to calls to FindCaption, FindData, //GetNearestItem, etc. It also gets	then Exit;
called for each keystroke sent to the	with ShellItem(Item.Index)^ do
//ListView (for incremental	begin
searching)	Item.Caption := DisplayName;
var	Item.ImageIndex := ImageIndex;
I: Integer;	Tremmingernaex: Imagemaex,
Found: Boolean;	if ListView.ViewStyle <> vsReport
begin	then Exit;
I := StartIndex;	their Daity
if (Find = ifExactString) or (Find =	if not IsFolder(FIShellFolder, ID)
ifPartialString) then	then
begin	VALVII
repeat	Item.SubItems.Add(Format('%dKB
if (I = FIDList.Count-1) then	', [Size]))
if Wrap then I := 0 else Exit;	else
Found :=	Item.SubItems.Add('');
Pos(UpperCase(FindString),	Item.SubItems.Add(TypeName);
UpperCase(ShellItem(I)^.DisplayNa	try
me)) = 1;	Item.SubItems.Add(ModDate);
Inc(I);	except
until Found or (I = StartIndex);	end;
if Found then Index := I-1;	· · ·
end;	if Bool(Attributes and
end;	FILE ATTRIBUTE READONLY)
	then Attrs := Attrs + 'R';
<> procedure	if Bool(Attributes and
TForm1.ListViewCustomDrawItem(	FILE ATTRIBUTE HIDDEN) then
Sender: TCustomListView;	Attrs := Attrs + 'H';
Item: TListItem; State:	if Bool(Attributes and
TCustomDrawState; var	FILE_ATTRIBUTE_SYSTEM) then
DefaultDraw: Boolean);	Attrs := Attrs + 'S';
var	if Bool(Attributes and
Attrs: Integer;	FILE_ATTRIBUTE_ARCHIVE)
begin	then Attrs := Attrs + 'A';
if Item = nil then Exit;	end;
Attrs :=	Item.SubItems.Add(Attrs);
ShellItem(Item.Index).Attributes;	end;
if Bool(Attrs and	
FILE ATTRIBUTE READONLY)	<> procedure
then	TForm1.ListViewDataFind(Sender:
ListView.Canvas.Font.Color :=	TObject; Find: TItemFind;
clGrayText;	const FindString: String; const
* '	

	if Bool(Attrs and
<> object \$14	FILE_ATTRIBUTE_HIDDEN) then
الملف النصي لبرنامج	ListView.Canvas.Font.Style :=
	ListView.Canvas.Font.Style +
Virtual Listview)	[fsStrikeOut];
عامدا	if Bool(Attrs and FILE_ATTRIBUTE_SYSTEM) then
علاء الدين اللباد	Listview.Canvas.Font.Color := clHighlight;
0.7.70.10.0011.11	end;
هاتف ۲۷۵۷۵۳۷۱	
<> object Form1: TForm1	<> procedure
Left = 174	TForm1.ListViewCustomDrawSubIt
Top = 113	em(Sender: TCustomListView;
Width = 569	Item: TListItem; SubItem: Integer;
Height = 448	State: TCustomDrawState;
Caption = 'Virtual List View Demo'	var DefaultDraw: Boolean);
Color = clBtnFace	begin
ParentFont = True	if SubItem = 0 then Exit;
OldCreateOrder = True	ListView.Canvas.Font.Color :=
Position = poScreenCenter	GetSysColor(COLOR_WINDOWT
Visible = True	EXT);
OnClose = Form1Close	//workaround for Win98 bug.
OnCreate = FormCreate	end;
PixelsPerInch = 96	,
TextHeight = 13	<> procedure
<> object ListView:	TForm1.btnBackClick(Sender:
TListView	TObject);
Left = 0	var
	Temp: PItemIDList;
Top = 56 Width = 561	begin
	Temp := CopyPIDL(FPIDL);
Height = 358	if Assigned(Temp) then
Align = alClient Columns = <	StripLastID(Temp);
	if Temp.mkid.cb <> 0 then
item	SetPath(Temp)
Caption = 'Name' Width = 175	else
	Beep;
end	end;
item	
Alignment = taRightJustify	<> procedure
Caption = 'Size'	TForm1.Form1Close(Sender:
Width = 75	TObject; var Action: TCloseAction);
end	begin
item	ClearIDList;
Caption = 'Type'	FIDList.Free;
Width = 150	end;
end	*****
item	end.
	Circle

AutoSize = True
ButtonHeight = 27
ButtonWidth = 28
Caption = 'ToolBar2'
Flat = True
Images = ToolbarImages
TabOrder = 0
<> object btnBrowse:
TToolButton
Left = 0
Top = 0
Caption = 'btnBrowse'
ImageIndex = 0
OnClick = btnBrowseClick
end
<> object btnBack:
TToolButton
Left = 28
Top = 0
Caption = 'btnBack'
ImageIndex = 5
OnClick = btnBackClick
end
<> object ToolButton3:
TToolButton
Left = 56
Top = 0
Width = 8
Caption = 'ToolButton3'
ImageIndex = 5
Style = tbsSeparator
end
<> object btnLargeIcons:
TToolButton
Left = 64
Top = 0
Caption = 'btnLargeIcons'
Grouped = True
ImageIndex = 1
Style = tbsCheck
OnClick = btnLargeIconsClick
end
<> object btnSmallIcons:
TToolButton
Tag = 1
Left = 92
Top = 0
Caption = 'btnSmallIcons'

Caption = 'Modified'
Width = 100
end
item
Alignment = taRightJustify
Caption = 'Attributes'
Width = 75
end>
OwnerData = True
ReadOnly = True
TabOrder = 0
ViewStyle = vsReport
OnCustomDrawItem =
ListViewCustomDrawItem
OnCustomDrawSubItem =
ListViewCustomDrawSubItem
OnData = ListViewData
OnDataFind = ListViewDataFind
OnDataHint = ListViewDataHint
OnDblClick = ListViewDblClick
OnKeyDown = ListViewKeyDown
end
<> object CoolBar1:
TCoolBar
Left = 0
Top = 0
Width = 561
Height = 56
AutoSize = True
Bands = <
item
Control = ToolBar2
ImageIndex = -1
MinHeight = 29
Width = 557
end
item
Control = cbPath
ImageIndex = -1
MinHeight = 21
Width = 557
end>
<> object ToolBar2:
TToolBar
Left = 9
Top = 0
Width = 544
Height = 29

000000000000000000000000000000000000000
end
<> object PopupMenu1:
TPopupMenu
Left = 160
Top = 128
end
end

C 1 T
Grouped = True
ImageIndex = 2
Style = tbsCheck
OnClick = btnLargeIconsClick
end
<> object btnList:
TToolButton
Tag = 2
Left = 120
Top = 0
Caption = 'btnList'
Grouped = True
ImageIndex = 3
Style = tbsCheck
OnClick = btnLargeIconsClick
end
<> object btnReport:
TToolButton
Tag = 3
Left = 148
Top = 0
Caption = 'btnReport'
Grouped = True
ImageIndex = 4
Style = tbsCheck
OnClick = btnLargeIconsClick
end
end
<> object cbPath:
TComboBox
Left = 9
Top = 31
Width = 544
Height = 21
ItemHeight = 13
TabOrder = 1
OnClick = cbPathClick
OnKeyDown = cbPathKeyDown
end
end
<> object ToolbarImages:
TImageList
Height = 21
Width = 21
Left = 72
Top = 128
-
Bitmap = {

Application.CreateForm(TFileAttrForm);
Application.CreateForm(TChangeDl g, ChangeDlg);
Application.Run;
end.
\\\\\\\\\
unit FChngDlg;
W. W.
interface
W. 1 0 W. 2
uses Windows, SysUtils, Classes,
Graphics, Forms, Controls, StdCtrls,
Buttons, ExtCtrls;
type
TChangeDlg = class(TForm)
OKBtn: TButton;
CancelBtn: TButton;
Bevel1: TBevel;
CurrentDir: TLabel;
Label1: TLabel;
Label2: TLabel;
ToFileName: TEdit;
FromFileName: TEdit;
private
{ Private declarations } public
{ Public declarations }
end;
var
ChangeDlg: TChangeDlg;
implementation
(00.1.10.)
{\$R *.dfm}
and
end.
\\\\\\\\\\
unit FAttrDlg;

الملف التنفيذي لبرنامج
34 procedures
Filmanex)
إعداد
علاء الدين اللباد
واتف ۷۹۴۵۷۷۳۷۱
program FilManEx;
uses
Forms,
FMXWin in 'FMXWin.pas'
{FMForm},
FmxUtils in 'Fmxutils.pas', FAttrDlg in 'FAttrDlg.pas'
{FileAttrForm},
FChngDlg in 'FChngDlg.pas'
{ChangeDlg};
(Change Dig),
{\$R *.RES}
begin
Application.Initialize;
Application.CreateForm(TFMForm, FMForm);

StatusBar: TPanel;
DirectoryPanel: TPanel;
FilePanel: TPanel;
DriveTabSet: TTabSet;
DirectoryOutline:
TDirectoryOutline;
FileList: TFileListBox;
MainMenu1: TMainMenu;
File1: TMenuItem;
Open1: TMenuItem;
Movel: TMenuItem;
Copy1: TMenuItem;
Delete1: TMenuItem;
Rename1: TMenuItem;
Properties1: TMenuItem;
N1: TMenuItem;
Exit1: TMenuItem;
Floppy: TImage;
Fixed: TImage;
Network: TImage;
CDRom: TImage;
RamDisk: TImage;
<> procedure
Exit1Click(Sender: TObject);
<> procedure
FormCreate(Sender: TObject);
<> procedure
DirectoryOutlineChange(Sender:
TObject);
<> procedure
FileListChange(Sender: TObject);
<> procedure
DriveTabSetMeasureTab(Sender:
TObject; Index: Integer;
var TabWidth: Integer);
<> procedure
DriveTabSetDrawTab(Sender:
TObject; TabCanvas: TCanvas;
R: TRect; Index: Integer;
Selected: Boolean);
<> procedure
File1Click(Sender: TObject);
<> procedure
Delete1Click(Sender: TObject);
<> procedure
Properties1Click(Sender: TObject);
<> procedure
FileChange(Sender: TObject);
<> procedure
protonate

• , e
interface
W. 1 2 W. 1 22
uses Windows, SysUtils, Classes,
Graphics, Forms, Controls, StdCtrls,
Buttons, ExtCtrls;
type
TFileAttrForm = class(TForm)
OKBtn: TButton;
CancelBtn: TButton;
Bevel1: TBevel;
FileDirName: TLabel;
FilePathName: TLabel;
ChangeDate: TLabel;
GroupBox1: TGroupBox;
ReadOnly: TCheckBox;
Archive: TCheckBox;
System: TCheckBox;
Hidden: TCheckBox;
private
{ Private declarations }
public
{ Public declarations }
end;
way
var
1.55
FileAttrForm: TFileAttrForm;
FileAttrForm: TFileAttrForm;
1.55
FileAttrForm: TFileAttrForm; implementation
FileAttrForm: TFileAttrForm;
FileAttrForm: TFileAttrForm; implementation  {\$R *.dfm}
FileAttrForm: TFileAttrForm; implementation
FileAttrForm: TFileAttrForm; implementation  {\$R *.dfm} end.
FileAttrForm: TFileAttrForm;  implementation  {\$R *.dfm}  end.
FileAttrForm: TFileAttrForm; implementation  {\$R *.dfm} end.
FileAttrForm: TFileAttrForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
FileAttrForm: TFileAttrForm;  implementation  {\$R *.dfm}  end.
FileAttrForm: TFileAttrForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
FileAttrForm: TFileAttrForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
FileAttrForm: TFileAttrForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
FileAttrForm: TFileAttrForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
FileAttrForm: TFileAttrForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
FileAttrForm: TFileAttrForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
FileAttrForm: TFileAttrForm;  implementation  {\$R *.dfm}  end.                       unit FMXWin;  interface  uses  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
FileAttrForm: TFileAttrForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
FileAttrForm: TFileAttrForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
FileAttrForm: TFileAttrForm;  implementation  {\$R *.dfm}  end.  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

way	Open1Click(Condon TObicat)
var	Open1Click(Sender: TObject);
Drive: Char;	<> procedure
AddedIndex: Integer;	FileListMouseDown(Sender:
begin	TObject; Button: TMouseButton;
for Drive := 'a' to 'z' do	Shift: TShiftState; X, Y:
begin	Integer);
case GetDriveType(PChar(Drive	<> procedure
+ ':\')) of	DirectoryOutlineDragOver(Sender,
DRIVE_REMOVABLE:	Source: TObject; X,
AddedIndex :=	Y: Integer; State: TDragState;
DriveTabSet.Tabs.AddObject(Drive,	var Accept: Boolean);
Floppy.Picture.Graphic);	<> procedure
DRIVE_FIXED:	DirectoryOutlineDragDrop(Sender,
AddedIndex :=	Source: TObject; X,
DriveTabSet.Tabs.AddObject(Drive,	Y: Integer);
Fixed.Picture.Graphic);	<> procedure
DRIVE_CDROM:	FileListEndDrag(Sender, Target:
AddedIndex :=	TObject; X, Y: Integer);
DriveTabSet.Tabs.AddObject(Drive,	<> procedure
CDRom.Picture.Graphic);	DriveTabSetChange(Sender:
DRIVE RAMDISK:	TObject; NewTab: Integer;
AddedIndex :=	var AllowChange: Boolean);
DriveTabSet.Tabs.AddObject(Drive,	private
RamDisk.Picture.Graphic);	<> procedure
DRIVE REMOTE:	ConfirmChange(const ACaption,
AddedIndex :=	FromFile, ToFile: string);
DriveTabSet.Tabs.AddObject(Drive,	public
Network.Picture.Graphic);	{ Public declarations }
else	end;
AddedIndex := 0;	
end;	var
if UpCase(Drive) = FileList.Drive	FMForm: TFMForm;
then	
DriveTabSet.TabIndex :=	implementation
AddedIndex;	
end;	uses FmxUtils, FAttrDlg, FChngDlg;
end;	
	{\$R *.dfm}
<> procedure	
TFMForm.DriveTabSetChange(Sen	<> procedure
der: TObject; NewTab: Integer;	TFMForm.Exit1Click(Sender:
var AllowChange: Boolean);	TObject);
begin	begin
if not (csDesigning in	Close;
ComponentState) then	end;
begin	
AllowChange := True;	<> procedure
try	TFMForm.FormCreate(Sender:
with DriveTabSet do	TObject);
	<u> </u>

var TabWidth: Integer);	DirectoryOutline.Drive :=
	Tabs[NewTab][1];
PitmanWidth, Integral	except
BitmapWidth: Integer;	on EInOutError do
begin DitmonWidth	
BitmapWidth := TBitmap(DriveTabSat Taba Objects	begin
TBitmap(DriveTabSet.Tabs.Objects	AllowChange := False; with DriveTabSet do
[Index]).Width;	
Inc(TabWidth, 2 + BitmapWidth);	DirectoryOutline.Drive :=
end;	Tabs[TabIndex][1];
Z nuo saduus	raise;
<> procedure	end;
TFMForm.DriveTabSetDrawTab(Se	end;
nder: TObject; TabCanvas:	end;
TCanvas;	end;
R: TRect; Index: Integer; Selected:	
Boolean);	<> procedure
Ditmon TDitmon	TFMForm.DirectoryOutlineChange(
Bitmap: TBitmap;	Sender: TObject);
begin	begin
Bitmap :=	FileList.Directory :=
TBitmap(DriveTabSet.Tabs.Objects	DirectoryOutline.Directory;
[Index]);	DirectoryPanel.Caption :=
with TabCanvas do	DirectoryOutline.Directory;
begin (D.L. 6) D.T 4 D.Y.	end;
Draw(R.Left, R.Top + 4, Bitmap);	
TextOut(R.Left + 2 +	<> procedure
Bitmap.Width, R.Top + 2,	TFMForm.FileListChange(Sender:
DriveTabSet.Tabs[Index]);	TObject);
end;	var
end;	TheFileName: string;
	begin
<> procedure	with FileList do
TFMForm.File1Click(Sender:	begin
TObject);	if ItemIndex >= 0 then
Var	begin
FileSelected: Boolean;	TheFileName :=
begin	Items[ItemIndex];
FileSelected := FileList.ItemIndex	FilePanel.Caption :=
>= 0;	Format('%s, %d bytes',
Open1.Enabled := FileSelected;	[TheFileName,
Delete1.Enabled := FileSelected;	GetFileSize(TheFileName)]);
Copy1.Enabled := FileSelected;	end
Move1.Enabled := FileSelected;	else FilePanel.Caption := ";
Rename1.Enabled := FileSelected;	end;
Properties1.Enabled :=	end;
FileSelected;	
end;	<> procedure
	TFMForm.DriveTabSetMeasureTab
<> procedure	(Sender: TObject; Index: Integer;

if System.Checked then	TFMForm.Delete1Click(Sender:
NewAttributes := NewAttributes or	TObject);
faSysFile	begin
else NewAttributes :=	with FileList do
NewAttributes and not faSysFile;	if MessageDlg('Delete ' +
if Hidden.Checked then	FileName + '?', mtConfirmation,
NewAttributes := NewAttributes or	[mbYes, mbNo], 0) = mrYes then
faHidden	if DeleteFile(FileName) then
else NewAttributes :=	Update;
NewAttributes and not faHidden;	end;
if NewAttributes <> Attributes	
then	<> procedure
	TFMForm.Properties1Click(Sender:
FileSetAttr(FileDirName.Caption,	TObject);
NewAttributes);	var
end;	Attributes, NewAttributes: Word;
end;	begin
end;	
Cita,	with FileAttrForm do
/ > nwoooduwo	begin
<> procedure	FileDirName.Caption :=
TFMForm.ConfirmChange(const	FileList.Items[FileList.ItemIndex];
ACaption, FromFile, ToFile: string);	FilePathName.Caption :=
begin	FileList.Directory;
if MessageDlg(Format('%s %s to	ChangeDate.Caption :=
%s?', [ACaption, FromFile,	DateTimeToStr(FileDateTime(FileLi
ToFile]),	st.FileName));
mtConfirmation, [mbYes, mbNo],	Attributes :=
0) = mrYes then	FileGetAttr(FileDirName.Caption);
begin	ReadOnly.Checked := (Attributes
if ACaption = 'Move' then	and faReadOnly) = faReadOnly;
MoveFile(FromFile, ToFile)	Archive.Checked := (Attributes
else if ACaption = 'Copy' then	and faArchive) = faArchive;
CopyFile(FromFile, ToFile)	System.Checked := (Attributes
else if ACaption = 'Rename' then	and faSysFile) = faSysFile;
RenameFile(FromFile, ToFile);	<b>Hidden.Checked := (Attributes</b>
FileList.Update;	and faHidden) = faHidden;
end;	if ShowModal <> mrCancel then
end;	begin
	NewAttributes := Attributes;
<> procedure	if ReadOnly.Checked then
TFMForm.FileChange(Sender:	<b>NewAttributes</b> := <b>NewAttributes</b> or
TObject);	faReadOnly
begin	else NewAttributes :=
with ChangeDlg do	NewAttributes and not faReadOnly;
begin	if Archive.Checked then
if Sender = Move1 then Caption :=	NewAttributes := NewAttributes or
'Move'	faArchive
else if Sender = Copy1 then	else NewAttributes :=
Caption := 'Copy'	NewAttributes and not faArchive;
Сарион Сору	THE WILLIAM IN THE HOLL HAVE CHIEFE,

Y: Integer; State: TDragState; var
Accept: Boolean);
begin
Accept := (Source is TFileListBox)
and (DirectoryOutline.GetItem(X,
(Y) > 0;
end;
<> procedure
TFMForm.DirectoryOutlineDragDr
op(Sender, Source: TObject; X,
Y: Integer);
begin
if Source is TFileListBox then
with DirectoryOutline do
ConfirmChange('Move',
FileList.FileName, Items[GetItem(X,
Y)].FullPath);
end;
<> procedure
TFMForm.FileListEndDrag(Sender,
Target: TObject; X, Y: Integer);
begin
if Target <> nil then
FileList.Update;
end;
end.
111111111111111111111111111111111111111

_		
•	<> object \$43	
	الهلف النصي لبرناهج	
	Filmanex)	
4	إعداد	
4	علاء الدين اللباد	
	واتف ۹۶٤٥٧٥٣٧١٠	
<	<> object FMForm: TFMForm	
Left = -4		
<b>Top = -4</b>		
1	Width = 1032	
]	Height = 746	
(	Caption = 'File manager example'	

else if Sender = Rename1 then
Caption := 'Rename'
else Exit;
CurrentDir.Caption :=
DirectoryOutline.Directory; FromFileName.Text :=
From FileName. Text := FileList.FileName;
ToFileName.Text := ";
if (ShowModal <> mrCancel) and
(ToFileName.Text $>$ ") then
ConfirmChange(Caption,
FromFileName.Text,
ToFileName.Text);
end;
end;
,
<> procedure
TFMForm.Open1Click(Sender:
TObject);
begin
with FileList do
begin
if HasAttr(FileName, faDirectory)
then
DirectoryOutline.Directory :=
FileName
else ExecuteFile(FileName, '',
Directory, SW_SHOW);
end;
end;
<> procedure
TFMForm.FileListMouseDown(Sen
der: TObject; Button:
TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin
if Button = mbLeft then
with Sender as TFileListBox do
begin  if Itom At Pas (Paint (Y. V), True)
if ItemAtPos(Point(X, Y), True) >= 0 then
BeginDrag(False);
end;
end;
ciiu,
<> procedure
TFMForm.DirectoryOutlineDragOv
er(Sender, Source: TObject; X,
(~ Jinati, Starter, 1 Object, 11,

Visible = False
end
<> object Fixed: TImage
Left = 48
Top = 16
Width = 16
Height = 9
AutoSize = True
Picture.Data = {
Visible = False
end
<> object Network: TImage
Left = 72
Top = 16
Width = 16
Height = 9
AutoSize = True
Picture.Data = {
Visible = False
end
<> object CDRom: TImage
Left = 98
Top = 16
<b>Width = 16</b>
Height = 9
AutoSize = True
Picture.Data = {
Visible = False
end
<> object RamDisk:
TImage
Left = 118
Top = 16
Width = 16
Height = 9
AutoSize = True
Picture.Data = {
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
Visible = False
end
end
end
<> object DriveTabSet:
TTabSet
Left = 0
Top = 631
Width = 1024

G I ID. E
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
Menu = MainMenu1
OldCreateOrder = True
Position = poDefault
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object StatusBar: TPanel
Left = 0
Top = 652
Width = 1024
Height = 40
Align = alBottom
BevelOuter = bvNone
TabOrder = 0
<> object DirectoryPanel:
TPanel
Left = 0
Top = 0
Width = 185
Height = 40
Align = alLeft
BevelInner = bvLowered
BevelWidth = 2
TabOrder = 0
end
<> object FilePanel: TPanel
Left = 185
Top = 0
Width = 839
Height = 40
Align = alClient
BevelInner = bvLowered
BevelWidth = 2
TabOrder = 1
<> object Floppy: TImage
Left = 24
Top = 16 Width = 16
Height = 9
AutoSize = True
Picture.Data = {

OnMouseDown =
FileListMouseDown
end
<> object MainMenu1:
TMainMenu
Left = 337
Top = 251
<> object File1: TMenuItem
Caption = '&File'
OnClick = File1Click
<> object Open1:
TMenuItem
Caption = '&Open'
ShortCut = 13
OnClick = Open1Click
end
<> object Move1:
TMenuItem
Caption = '&Move'
ShortCut = 118
OnClick = FileChange
end
<> object Copy1:
TMenuItem
Caption = '&Copy'
ShortCut = 119
OnClick = FileChange
end
<> object Delete1:
TMenuItem
Caption = '&Delete'
ShortCut = 46
OnClick = Delete1Click
end
<> object Rename1:
TMenuItem
Caption = '&Rename'
OnClick = FileChange
end
<> object Properties1:
TMenuItem
Caption = '&Properties'
ShortCut = 32781
OnClick = Properties1Click
end
<> object N1: TMenuItem
Caption = '-'
end

Height = 21
Align = alBottom
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -12
Font.Name = 'MS Sans Serif'
Font.Style = []
Style = tsOwnerDraw
OnChange = DriveTabSetChange
OnDrawTab =
DriveTabSetDrawTab
OnMeasureTab =
DriveTabSetMeasureTab
end
<> object DirectoryOutline:
<b>TDirectoryOutline</b>
Left = 0
Top = 0
Width = 161
Height = 631
Align = alLeft
ItemHeight = 13
Options = [ooDrawFocusRect]
PictureLeaf.Data = {
TabOrder = 2
OnChange =
DirectoryOutlineChange
OnDragDrop =
DirectoryOutlineDragDrop
OnDragOver =
DirectoryOutlineDragOver
Data = {10}
end
<> object FileList:
TFileListBox
Left = 161
Top = 0
Width = 863
Height = 631
Align = alClient
FileType = [ftReadOnly,
ftDirectory, ftArchive, ftNormal]
ItemHeight = 16
ShowGlyphs = True
TabOrder = 3
OnChange = FileListChange
OnEndDrag = FileListEndDrag
January I mention in

end
<> object ChangeDate:
TLabel
Left = 24
Top = 64
Width = 60
Height = 13
Caption = 'ChangeDate'
end
<> object OKBtn: TButton
Left = 300
Top = 8
Width = 77
Height = 27
Caption = 'OK'
Default = True
ModalResult = 1
TabOrder = 0
end
<> object CancelBtn: TButton
Left = 300
Top = 40
<b>Width</b> = 77
Height = 27
Cancel = True
Caption = 'Cancel'
ModalResult = 2
TabOrder = 1
end
<> object GroupBox1:
TGroupBox
Left = 24
Top = 88
Width = 249
Height = 65
Caption = 'Attributes'
TabOrder = 2
<> object ReadOnly:
TCheckBox
Left = 16
Top = 16
Width = 97
Height = 17
Caption = '&Read Only'
TabOrder = 0
end
<> object Archive:
TCheckBox

A shippet Evit1.
<> object Exit1:
TMenuItem
Caption = 'E&xit'
OnClick = Exit1Click
end
end
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
<> object FileAttrForm:
TFileAttrForm
Left = 180
Top = 118
ActiveControl = OKBtn
BorderStyle = bsDialog
· ·
Client Height - 170
ClientHeight = 179
ClientWidth = 388
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
PixelsPerInch = 96
TextHeight = 13
<> object Bevel1: TBevel
Left = 8
Top = 8
Width = 281
Height = 161
Shape = bsFrame
IsControl = True
end
<> object FileDirName:
TLabel
Left = 24
Top = 16 Width = 57
Width = 57
Height = 13
Caption = 'FileDirName'
end
<> object FilePathName:
TLabel
Left = 24
Top = 40
Width = 66
Height = 13
Caption = 'FilePathName'
-

Top = 8
Width = 281
Height = 105
Shape = bsFrame
IsControl = True
end
<> object CurrentDir: TLabel
Left = 24
Top = 24
Width = 47
Height = 13
Caption = 'CurrentDir'
end
<> object Label1: TLabel
Left = 24
Top = 48
Width = 26
Height = 13
Caption = 'From:'
end
<> object Label2: TLabel
Left = 24
Top = 80
Width = 16
Height = 13
Caption = 'To:'
end
<> object OKBtn: TButton
Left = 300
Top = 8
Width = 77
Height = 27
Caption = 'OK'
Default = True
ModalResult = 1
TabOrder = 0
end
<> object CancelBtn: TButton
Left = 300
Top = 40
Width = 77
Height = 27
Cancel = True
Caption = 'Cancel'
ModalResult = 2
TabOrder = 1
end
<> object ToFileName: TEdit
object for hervalle. Thul

Left = 16
Top = 40
<b>Width = 97</b>
Height = 17
Caption = '&Archive'
TabOrder = 1
end
<> object System:
TCheckBox
Left = 120
Top = 16
Width = 97
Height = 17
Caption = '&System'
TabOrder = 2
end
<> object Hidden:
TCheckBox
Left = 120
Top = 40
Width = 97
Height = 17
Caption = '&Hidden'
TabOrder = 3
end
end
End
Enu
1_1
\ <u>=</u> \
111111111111111111111111111111111111111
<> object ChangeDlg:
TChangeDlg
Left = 188
Top = 338
ActiveControl = OKBtn
BorderStyle = bsDialog
Caption = 'Dialog'
ClientHeight = 123
ClientWidth = 388
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
PixelsPerInch = 96
TextHeight = 13
<> object Bevel1: TBevel
Left = 8

DVMisa HayDumn
RXMisc, HexDump,
ImgList;
type
TMainForm = class(TForm)
StatusBar: TStatusBar;
TreeViewPanel: TPanel;
Panel1: TPanel;
ImageViewer: TImage;
ListView: TListView;
TreeView: TTreeView;
Splitter: TPanel;
Notebook: TNotebook;
ListViewPanel: TPanel;
ListViewCaption: TPanel;
FileOpenDialog: TOpenDialog;
FileSaveDialog: TSaveDialog;
MainMenu: TMainMenu;
miFile: TMenuItem;
miFileExit: TMenuItem;
miFileSave: TMenuItem;
miFileOpen: TMenuItem;
miView: TMenuItem;
miViewStatusBar: TMenuItem;
miViewLargeIcons: TMenuItem;
miViewSmallIcons: TMenuItem;
miViewList: TMenuItem;
miViewDetails: TMenuItem;
miHelp: TMenuItem;
miHelpAbout: TMenuItem;
Small: TImageList;
Large: TImageList;
StringViewer: TMemo;
<> procedure
FileExit(Sender: TObject);
<> procedure
FileOpen(Sender: TObject);
<> procedure
FormCreate(Sender: TObject);
<> procedure
ListViewEnter(Sender: TObject);
<> procedure
SaveResource(Sender: TObject);
<> procedure
SelectListViewType(Sender:
TObject);
<> procedure
ShowAboutBox(Sender: TObject);
ShowAboutDox(Sender: 1 Object);

Left = 64
Top = 80
Width = 209
Height = 21
TabOrder = 2
end
<> object FromFileName:
TEdit
Left = 64
Top = 48
Width = 209
Height = 21
TabOrder = 3
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

الملف التنفيذي لبرنامج
42 procedures
ResXplor)
إعداد
علاء الدين اللباد
واتف ۷۱۳۵۷۵۶۹۰
unit RXMain;
{ This program provides an example
of how to use the TreeView and
ListView
components in a fashion similar to
the Microsoft Windows Explorer.
It is not intended to be a fully
functional resource viewer. }
interface
uses
SysUtils, Windows, Messages,
Classes, Graphics, Controls, Forms,
Dialogs,
ExeImage, StdCtrls, Buttons,

ExtCtrls, ComCtrls, Menus,

	<> procedure
uses About, RXTypes;	ToggleStatusBar(Sender: TObject);
	<> procedure
{\$R *.dfm}	TreeViewChange(Sender: TObject;
{\$R RXIMAGES.RES}	Node: TTreeNode);
	<> procedure
const	SplitterMouseDown(Sender:
itBitmap: TResType =	TObject; Button: TMouseButton;
ImgList.rtBitmap; // Reference for	Shift: TShiftState; X, Y:
duplicate identifier	Integer);
ImageMap: array[TResourceType]	<> procedure
of Byte =	SplitterMouseMove(Sender:
(2,4,5,3,2,2,2,2,2,2,2,2,2,2,2,2,2);	TObject; Shift: TShiftState; X,
ResFiltMap:	Y: Integer);
array[TResourceType] of Byte =	<> procedure
(1,4,2,3,1,1,1,1,1,1,1,1,1,1,1,1);	SplitterMouseUp(Sender: TObject;
	<b>Button: TMouseButton;</b>
SNoResSelected = 'No resource	Shift: TShiftState; X, Y:
selected';	Integer);
SFormCaption = 'Resource	<> procedure
Explorer';	ViewMenuDropDown(Sender:
SSaveFilter = 'Other Resource	TObject);
(*.*) *.* Bitmaps (*.BMP) *.BMP '+	<> procedure
'Icons	NotebookEnter(Sender: TObject);
(*.ICO) *.ICO Cursor	<> procedure
(*.CUR) *.CUR';	FormDestroy(Sender: TObject);
SOpenFilter = 'Executable File	private
Images	FExeFile: TExeImage;
(*.EXE;*.DLL) *.EXE;*.DLL '+	HexDump: THexDump;
'All Files (*.*) *.*';	SplitControl: TSplitControl;
	<> procedure
{ Utility Functions }	LoadResources(ResList:
	TResourceList; Node: TTreeNode);
<> procedure Error(const	<> procedure
ErrMsg: string);	DisplayResources;
begin	<> procedure
raise Exception.Create(ErrMsg);	UpdateViewPanel;
end;	<> procedure
	UpdateListView(ResList:
<> procedure ErrorFmt(const	TResourceList);
ErrMsg: string; Params: array of	<> procedure
const);	UpdateStatusLine(ResItem:
begin	TResourceItem);
raise	end;
Exception.Create(format(ErrMsg,	
Params));	var
end;	MainForm: TMainForm;
function Confirm(const AMsg:	implementation

FileName)]);
with TreeView do
begin
SetFocus;
Selected := Items[0];
end;
end;
<> procedure
TMainForm.UpdateViewPanel;
var
R: TResourceItem;
begin
with TreeView do
begin
if Visible and Assigned(Selected)
then
begin
R :=
TResourceItem(Selected.Data);
if R.IsList then
UpdateListView(R.List) else
begin
case R.ResType of
rtBitmap, rtIconEntry,
rtCursorEntry:
begin
ImageViewer.Picture.Assign(R);
Notebook.PageIndex := 1;
end;
rtString, rtMenu:
begin
Saving Viewen Lines Assign (D)
StringViewer.Lines.Assign(R);
StringViewer.SelStart := 0;
Notebook.PageIndex := 2;
end
else
begin
HexDump.Address :=
R.RawData;
HexDump.DataSize :=
R.Size;
Notebook.PageIndex := 3;
end;
end;
end;

String): Boolean;
begin
Result := MessageDlg(AMsg,
mtConfirmation, mbYesNoCancel,
0) = idYes;
end;
{ Non Event Handlers }
(100 Event Handlers)
<> procedure
TMainForm.LoadResources(ResList
: TResourceList; Node: TTreeNode);
var L.
I: Integer;
CNode: TTreeNode;
begin
for I := 0 to ResList.Count - 1 do
with ResList[I] do
begin
CNode :=
TreeView.Items.AddChildObject(No
de, Name, ResList[I]);
if IsList then
begin
CNode.SelectedIndex := 1;
LoadResources(List, CNode);
end else
begin
CNode.ImageIndex :=
ImageMap[ResList[I].ResType];
CNode.SelectedIndex :=
CNode.ImageIndex;
end;
Ciiu,
ou de
end;
end;
<> procedure
TMainForm.DisplayResources;
begin
ListView.Items.Clear;
TreeView.Selected := nil;
TreeView.Items.Clear;
,
LoadResources(FExeFile.Resources,
nil);
Caption := Format('%s - %s',
[SFormCaption,
AnsiLowerCaseFileName(FExeFile.
THE THE TOTAL CASE HE WITHOUT TEACHING

StatusBar.Panels[1].Text :=	
Format(' Offset: %x Size: %x',	_ (
[Offset, Size]);	eı
end;	enc
end;	
	<
{ Form Initialization }	TN
	t: ]
<> procedure	vai
TMainForm.FormCreate(Sender:	I:
TObject);	beg
begin	L
SplitControl :=	fo
TSplitControl.Create(Self);	
HexDump :=	Lis
CreateHexDump(TWinControl(Note	
Book.Pages.Objects[3]));	
FileOpenDialog.Filter :=	
SOpenFilter;	
FileSaveDialog.Filter :=	[0]
SSaveFilter;	
Small.ResourceLoad(itBitmap,	[Si
'SmallImages', clOlive);	
Large.ResourceLoad(itBitmap,	Im
'LargeImages', clOlive);	(
Notebook.PageIndex := 0;	N
if (ParamCount > 0) and	ene
FileExists(ParamStr(1)) then	
begin	<
Show;	TN
FExeFile :=	ter
TExeImage.CreateImage(Self,	be
ParamStr(1));	if
DisplayResources;	b
end;	
end;	<b>'+</b> ]
{ Menu Event Handlers }	Fo
, , , , , , , , , , , , , , , , , , , ,	Li
<> procedure	
TMainForm.FileOpen(Sender:	Fo
TObject);	
var	[Re
TmpExeFile: TExeImage;	eı
	W
begin with FileOpen Dialog de	b
with FileOpenDialog do	
begin	Fo
if not Execute then Exit;	Na
TmpExeFile :=	

UndataStatusLina(D).
UpdateStatusLine(R);
end;
end;
end;
<> procedure
TMainForm.UpdateListView(ResLis
t: TResourceList);
var
I: Integer;
begin
ListView.Items.Clear;
for I := 0 to ResList.Count-1 do
with ResList[I],
ListView.Items.Add do
begin Data Baskistilla
Data := ResList[I];
Caption := Name;
SubItems.Add(Format('%.7x',
[Offset]));
SubItems.Add(Format('%.5x',
[Size]));
ImageIndex :=
ImageMap[ResType];
end;
Notebook.PageIndex := 0;
end;
<> procedure
TMainForm.UpdateStatusLine(ResI
tem: TResourceItem);
begin
if ResItem.IsList then
begin
ListViewCaption.Caption := '
'+TreeView.Selected.Text;
StatusBar.Panels[0].Text :=
Format(' %d object(s)',
[ListView.Items.Count]);
StatusBar.Panels[1].Text :=
Format(' Offset: %x',
[ResItem.Offset]);
end else
with ResItem do
begin
ListViewCaption.Caption :=
Format(' %s: %s', [ResTypeStr,
Namel);
StatusBar.Panels[0].Text := ";
~ ····································

ata);	TExeImage.CreateImage(Self,
end;	FileName);
	if Assigned(FExeFile) then
function	FExeFile.Destroy;
ListViewResourceSelected: Boolean;	FExeFile := TmpExeFile;
begin	DisplayResources;
Result :=	end;
Assigned(ListView.Selected) and	end;
	,
Assigned(ListView.Selected.Data)	<> procedure
and	TMainForm.FileExit(Sender:
not	TObject);
TResourceItem(ListView.Selected.D	begin
ata).IsList;	Close;
if Result then ResItem :=	end;
TResourceItem(ListView.Selected.D	enu,
ata);	Z S mussadums
end;	<> procedure
enu,	TMainForm.ListViewEnter(Sender:
1 .	TObject);
begin	begin
if TreeViewResourceSelected or	with ListView do
ListViewResourceSelected then	if (Items.Count > 1) and (Selected
with FileSaveDialog do	= nil) then
begin	begin
FilterIndex :=	Selected := Items[0];
ResFiltMap[ResItem.ResType];	ItemFocused := Selected;
if Execute then	end;
ResItem.SaveToFile(FileName)	end;
end	
else	<> procedure
Error(SNoResSelected);	TMainForm.SaveResource(Sender:
end;	TObject);
	var
<> procedure	ResItem: TResourceitem;
TMainForm.SelectListViewType(Se	
nder: TObject);	function
begin	TreeViewResourceSelected:
ListView.ViewStyle :=	Boolean;
TViewStyle(TComponent(Sender).T	begin
ag);	Result :=
end;	
Ciiu,	Assigned(TreeView.Selected) and
< > procedure	Assigned (TreeView Selected Date)
<> procedure TMoinForm Show About Poy (Sondon)	Assigned(TreeView.Selected.Data)
TMainForm.ShowAboutBox(Sender	and
: TObject);	not
begin	TResourceItem(TreeView.Selected.D
About.ShowAboutBox;	ata).IsList;
end;	if Result then ResItem :=
	TResourceItem(TreeView.Selected.D

end;
<> procedure
TMainForm.SplitterMouseUp(Sende
r: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin
with SplitControl do if Sizing then
EndSizing;
end;
<> procedure
TMainForm.NotebookEnter(Sender:
TObject);
var
Page: TWinControl;
begin
with NoteBook do
begin
Page :=
TWinControl(Pages.Objects[PageIn
dex]);
if (Page.ControlCount > 0) and
(Page.Controls[0] is TWinControl)
then
TWinControl(Page.Controls[0]).Set
Focus;
end;
end;
<> procedure
TMainForm.FormDestroy(Sender:
TObject);
begin
SplitControl.Free;
end;
end.

<> object35	
الملف النصي لبرناهج	
ResXplor)	
إعداد	
علاء الدين اللباد	

<> procedure
TMainForm.ToggleStatusBar(Sende
r: TObject);
begin
StatusBar.Visible := not
StatusBar.Visible;
end;
enu,
<> procedure
TMainForm.TreeViewChange(Send
er: TObject; Node: TTreeNode);
begin
UpdateViewPanel;
end;
<> procedure
TMainForm.ViewMenuDropDown(
Sender: TObject);
var
I: Integer;
begin
miViewStatusBar.Checked :=
StatusBar.Visible;
for I := 0 to miView.Count-1 do
with miView.Items[I] do
if GroupIndex = 1 then
Checked := (Tag =
Ord(ListView.ViewStyle));
end;
<> procedure
TMainForm.SplitterMouseDown(Se
nder: TObject;
Button: TMouseButton; Shift:
TShiftState; X, Y: Integer);
begin
if (Button = mbLeft) and (Shift =
[ssLeft]) then
SplitControl.BeginSizing(Splitter,
TreeViewPanel);
end;
<> procedure
TMainForm.SplitterMouseMove(Se
nder: TObject; Shift: TShiftState;
X, Y: Integer);
begin
with SplitControl do if Sizing then
ChangeSizing(X, Y);
Change Ching(11) 1 J9

Alignment = taLeftJustify
BevelInner = bvLowered
BevelOuter = bvNone
BorderWidth = 1
Caption = ' Resources'
TabOrder = 0
end
<> object TreeView:
TTreeView
Left = 0
Top = 21
Width = 185
Height = 652
Align = alClient
Images = Small
Indent = 19
ReadOnly = True
TabOrder = 1
OnChange = TreeViewChange
end
end
<> object Splitter: TPanel
Left = 185
Top = 0
Width = 2
Height = 673
Cursor = crHSplit
Align = alLeft
BevelOuter = bvNone
TabOrder = 2
OnMouseDown =
SplitterMouseDown
OnMouseMove =
SplitterMouseMove
OnMouseUp = SplitterMouseUp
end
<> object ListViewPanel:
TPanel
Left = 187
Top = 0
Width = 837
Height = 673
Align = alClient
BevelOuter = bvNone
TabOrder = 3
<> object ListViewCaption:
TPanel
Left = 0

واتف ۷۹۲۵۷۷۴۱
<> object MainForm:
TMainForm
Left = -4
Top = -4
Width = 1032
Height = 746
Caption = 'Resource Explorer'
Color = clBtnFace
ParentFont = True
Icon.Data = {
0FF}
Menu = MainMenu
OldCreateOrder = True
OnCreate = FormCreate
OnDestroy = FormDestroy
PixelsPerInch = 96
TextHeight = 13
<> object StatusBar:
TStatusBar
Left = 0
Top = 673
Width = 1024
Height = 19
Panels = <
item
Width = 150
end
item
Width = 500
end>
end
<> object TreeViewPanel:
TPanel
Left = 0
Top = 0
Width = 185
Height = 673
Align = alLeft
BevelOuter = bvNone
TabOrder = 1
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 185
Height = 21
Align = alTop

SmallImages = Small TabOrder = 0 ViewStyle = vsReport OnEnter = ListViewEnter end end <> object TPage Left = 0 Top = 0 Caption = 'ImageViewPage' <> object ImageViewer: TImage Left = 0 Top = 0 Width = 837 Height = 652 Align = alClient Center = True end end <> object TPage Left = 0 Top = 0 Caption = 'StringViewPage' <> object StringViewer: TMemo Left = 0 Top = 0 Width = 382 Height = 175 Align = alClient ReadOnly = True ScrollBars = ssBoth TabOrder = 0
ViewStyle = vsReport OnEnter = ListViewEnter end end <
OnEnter = ListViewEnter end end <> object TPage Left = 0 Top = 0 Caption = 'ImageViewPage' <> object ImageViewer: TImage Left = 0 Top = 0 Width = 837 Height = 652 Align = alClient Center = True end end <> object TPage Left = 0 Top = 0 Caption = 'StringViewPage' <> object StringViewer: TMemo Left = 0 Top = 0 Width = 382 Height = 175 Align = alClient ReadOnly = True ScrollBars = ssBoth
end  <> object TPage  Left = 0  Top = 0  Caption = 'ImageViewPage'  <> object ImageViewer:  TImage  Left = 0  Top = 0  Width = 837  Height = 652  Align = alClient  Center = True  end  end  <> object TPage  Left = 0  Top = 0  Caption = 'StringViewPage'  <> object StringViewer:  TMemo  Left = 0  Top = 0  Width = 382  Height = 175  Align = alClient  ReadOnly = True  ScrollBars = ssBoth
end  <> object TPage  Left = 0  Top = 0  Caption = 'ImageViewPage'  <> object ImageViewer:  TImage  Left = 0  Top = 0  Width = 837  Height = 652  Align = alClient  Center = True  end end  <> object TPage  Left = 0  Top = 0  Caption = 'StringViewPage'  <> object StringViewer:  TMemo  Left = 0  Top = 0  Width = 382  Height = 175  Align = alClient  ReadOnly = True  ScrollBars = ssBoth
<pre>     Left = 0     Top = 0     Caption = 'ImageViewPage'     &lt;&gt; object ImageViewer: TImage     Left = 0     Top = 0     Width = 837     Height = 652     Align = alClient     Center = True     end     end     &lt;&gt; object TPage     Left = 0     Top = 0     Caption = 'StringViewPage'     &lt;&gt; object StringViewer: TMemo     Left = 0     Top = 0     Width = 382     Height = 175     Align = alClient     ReadOnly = True     ScrollBars = ssBoth </pre>
Left = 0 Top = 0 Caption = 'ImageViewPage' <> object ImageViewer: TImage  Left = 0 Top = 0 Width = 837 Height = 652 Align = alClient Center = True end end <> object TPage Left = 0 Top = 0 Caption = 'StringViewPage' <> object StringViewer: TMemo Left = 0 Top = 0 Width = 382 Height = 175 Align = alClient ReadOnly = True ScrollBars = ssBoth
Top = 0 Caption = 'ImageViewPage' <> object ImageViewer: TImage  Left = 0 Top = 0 Width = 837 Height = 652 Align = alClient Center = True end end <> object TPage Left = 0 Top = 0 Caption = 'StringViewPage' <> object StringViewer: TMemo Left = 0 Top = 0 Width = 382 Height = 175 Align = alClient ReadOnly = True ScrollBars = ssBoth
Caption = 'ImageViewPage'  <> object ImageViewer:  TImage  Left = 0  Top = 0  Width = 837  Height = 652  Align = alClient  Center = True  end  end  <> object TPage  Left = 0  Top = 0  Caption = 'StringViewPage'  <> object StringViewer:  TMemo  Left = 0  Top = 0  Width = 382  Height = 175  Align = alClient  ReadOnly = True  ScrollBars = ssBoth
<pre>&lt;&gt; object ImageViewer: TImage     Left = 0     Top = 0     Width = 837     Height = 652     Align = alClient     Center = True     end     end     &lt;&gt; object TPage     Left = 0     Top = 0     Caption = 'StringViewPage'     &lt;&gt; object StringViewer: TMemo     Left = 0     Top = 0     Width = 382     Height = 175     Align = alClient     ReadOnly = True     ScrollBars = ssBoth</pre>
TImage  Left = 0  Top = 0  Width = 837  Height = 652  Align = alClient  Center = True  end  end  <> object TPage  Left = 0  Top = 0  Caption = 'StringViewPage'  <> object StringViewer:  TMemo  Left = 0  Top = 0  Width = 382  Height = 175  Align = alClient  ReadOnly = True  ScrollBars = ssBoth
Left = 0 Top = 0 Width = 837 Height = 652 Align = alClient Center = True end end <> object TPage Left = 0 Top = 0 Caption = 'StringViewPage' <> object StringViewer: TMemo Left = 0 Top = 0 Width = 382 Height = 175 Align = alClient ReadOnly = True ScrollBars = ssBoth
Top = 0 Width = 837 Height = 652 Align = alClient Center = True end end <> object TPage Left = 0 Top = 0 Caption = 'StringViewPage' <> object StringViewer: TMemo Left = 0 Top = 0 Width = 382 Height = 175 Align = alClient ReadOnly = True ScrollBars = ssBoth
Width = 837 Height = 652 Align = alClient Center = True end end <> object TPage Left = 0 Top = 0 Caption = 'StringViewPage' <> object StringViewer: TMemo Left = 0 Top = 0 Width = 382 Height = 175 Align = alClient ReadOnly = True ScrollBars = ssBoth
Height = 652  Align = alClient  Center = True end end <> object TPage  Left = 0  Top = 0  Caption = 'StringViewPage' <> object StringViewer:  TMemo  Left = 0  Top = 0  Width = 382  Height = 175  Align = alClient  ReadOnly = True  ScrollBars = ssBoth
Align = alClient Center = True end end <> object TPage Left = 0 Top = 0 Caption = 'StringViewPage' <> object StringViewer: TMemo Left = 0 Top = 0 Width = 382 Height = 175 Align = alClient ReadOnly = True ScrollBars = ssBoth
Center = True end end <> object TPage Left = 0 Top = 0 Caption = 'StringViewPage' <> object StringViewer: TMemo Left = 0 Top = 0 Width = 382 Height = 175 Align = alClient ReadOnly = True ScrollBars = ssBoth
end end <> object TPage  Left = 0  Top = 0  Caption = 'StringViewPage' <> object StringViewer:  TMemo  Left = 0  Top = 0  Width = 382  Height = 175  Align = alClient  ReadOnly = True  ScrollBars = ssBoth
end <> object TPage  Left = 0  Top = 0  Caption = 'StringViewPage' <> object StringViewer:  TMemo  Left = 0  Top = 0  Width = 382  Height = 175  Align = alClient  ReadOnly = True  ScrollBars = ssBoth
<> object TPage  Left = 0  Top = 0  Caption = 'StringViewPage'  <> object StringViewer:  TMemo  Left = 0  Top = 0  Width = 382  Height = 175  Align = alClient  ReadOnly = True  ScrollBars = ssBoth
Left = 0 Top = 0 Caption = 'StringViewPage' <> object StringViewer: TMemo Left = 0 Top = 0 Width = 382 Height = 175 Align = alClient ReadOnly = True ScrollBars = ssBoth
Top = 0 Caption = 'StringViewPage' <> object StringViewer: TMemo Left = 0 Top = 0 Width = 382 Height = 175 Align = alClient ReadOnly = True ScrollBars = ssBoth
Caption = 'StringViewPage' <> object StringViewer: TMemo  Left = 0 Top = 0 Width = 382 Height = 175 Align = alClient ReadOnly = True ScrollBars = ssBoth
<> object StringViewer:  TMemo  Left = 0  Top = 0  Width = 382  Height = 175  Align = alClient  ReadOnly = True  ScrollBars = ssBoth
TMemo  Left = 0  Top = 0  Width = 382  Height = 175  Align = alClient  ReadOnly = True  ScrollBars = ssBoth
Left = 0 Top = 0 Width = 382 Height = 175 Align = alClient ReadOnly = True ScrollBars = ssBoth
Top = 0 Width = 382 Height = 175 Align = alClient ReadOnly = True ScrollBars = ssBoth
Width = 382 Height = 175 Align = alClient ReadOnly = True ScrollBars = ssBoth
Height = 175 Align = alClient ReadOnly = True ScrollBars = ssBoth
Align = alClient ReadOnly = True ScrollBars = ssBoth
ReadOnly = True ScrollBars = ssBoth
ScrollBars = ssBoth
TahOrdor - 0
Taboruer – v
WantReturns = False
WordWrap = False
end
end
<> object TPage
Left = 0
Top = 0
Caption = 'OtherViewPage'
end
end
end
<> object MainMenu:
TMainMenu

Top = 0
Width = 837
Height = 21
Align = alTop
Alignment = taLeftJustify
BevelInner = bvLowered
BevelOuter = bvNone
BorderWidth = 1
TabOrder = 0
end
<> object Notebook:
TNotebook
Left = 0
Top = 21
Width = 837
Height = 652
Align = alClient
Color = clBtnFace
PageIndex = 1
ParentColor = False
TabOrder = 1
OnEnter = NotebookEnter
<> object TPage
Left = 0
Top = 0
Caption = 'ListViewPage'
<> object ListView:
TListView
Left = 0
Top = 0
Width = 382
Height = 175
Align = alClient
Columns = <
item
Caption = 'Name'
Width = 150
end
item
Caption = 'Offset'
Width = 80
end
item
Caption = 'Size'
Width = 80
end>
ColumnClick = False
LargeImages = Large

GroupIndex = 1
RadioItem = True
OnClick = SelectListViewType
end
<> object miViewList:
TMenuItem
Tag = 2
Caption = '&List'
GroupIndex = 1
RadioItem = True
OnClick = SelectListViewType
end
<> object miViewDetails:
TMenuItem
Tag = 3
Caption = '&Details'
GroupIndex = 1
RadioItem = True
OnClick = SelectListViewType
end
end
<> object miHelp:
TMenuItem
Caption = '&Help'
<> object miHelpAbout:
TMenuItem
Caption = '&About'
OnClick = ShowAboutBox
end
end
end
<> object FileOpenDialog:
TOpenDialog
Options = [ofOverwritePrompt,
ofHideReadOnly, ofPathMustExist]
Left = 497
Top = 161
end
<> object FileSaveDialog:
TSaveDialog
Options = [ofOverwritePrompt,
ofHideReadOnly, ofPathMustExist]
Left = 463
Top = 161
end
<> object Small: TImageList
Left = 384
Top = 161

Left = 530
Top = 161
<> object miFile:
TMenuItem
Caption = '&File'
<> object miFileOpen:
TMenuItem
Caption = '&Open'
OnClick = FileOpen
end
<> object miFileSave:
TMenuItem
Caption = '&Save Resource'
OnClick = SaveResource
end
<> object miN1:
TMenuItem
Caption = '-'
end
<> object miFileExit:
TMenuItem
Caption = 'E&xit'
OnClick = FileExit
end
end
<> object miView:
TMenuItem
Caption = '&View'
OnClick = ViewMenuDropDown
<> object
miViewStatusBar: TMenuItem
Caption = 'Status &Bar'
OnClick = ToggleStatusBar
end
<> object N6: TMenuItem
Caption = '-'
end
<> object
miViewLargelcons: TMenuItem
Caption = 'Lar≥ Icons'
GroupIndex = 1
RadioItem = True
OnClick = SelectListViewType
end
<> object
miViewSmallIcons: TMenuItem
Tag = 1
Caption = 'S&mall Icons'

DBCtrls, StdCtrls, ExtCtrls, Mask,
DBTables, DB, Grids, DBGrids,
Menus,
DBCGrids;
type
TFmCtrlGrid = class(TForm)
DBCtrlGrid1: TDBCtrlGrid;
DBGrid1: TDBGrid;
DBEdit1: TDBEdit;
DBEdit2: TDBEdit;
DBEdit3: TDBEdit;
DBEdit4: TDBEdit;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
Button1: TButton;
Bevel1: TBevel;
Label8: TLabel;
Label9: TLabel;
Label10: TLabel;
Label11: TLabel;
lPurchase: TLabel;
lTotalCost: TLabel;
lTotalShares: TLabel;
IDifference: TLabel;
DBNavigator1: TDBNavigator;
MainMenu1: TMainMenu;
About1: TMenuItem;
<procedure< td=""></procedure<>
Button1Click(Sender: TObject);
<procedure< td=""></procedure<>
FormShow(Sender: TObject);
<procedure< td=""></procedure<>
DBGrid1Enter(Sender: TObject);
<procedure< td=""></procedure<>
DBCtrlGrid1Enter(Sender:
TObject);
<procedure< td=""></procedure<>
About1Click(Sender: TObject);
<procedure< td=""></procedure<>
FormCreate(Sender: TObject);
private
{ Private declarations }

end
<> object Large: TImageList
Height = 32
<b>Width = 32</b>
Left = 417
Top = 161
end
end

# القسم الخامس برمجة قواعد البيانات في لغة البرمجة دلفي

البرنامج يتعامل مع أكثر الإجراءات المتعلقة بقواعد البيانات وهو موجود في دلفي ٧ واسمه:

# **CtrlGrid**

من إعداد

علاء الدين اللباد ١

يحتوي البرنامج علىProcedure24
يتعامل مع قواعد البيانات
CtrlGridالملف التنفيذي للبرنامج للوحدة
unit CtrlForm;
interface
uses
Windows, Messages, SysUtils,
Classes Graphics Controls Forms

Dialogs,

ShowModal;	publ
finally	{ Pu
Free;	end;
end;	
end;	var
	FmC
<procedure< td=""><td></td></procedure<>	
TFmCtrlGrid.FormCreate(Sender:	imple
TObject);	
begin	uses I
end;	{\$R *.
	<p< td=""></p<>
end.	TFm(
	TObje
M	begin
aboutالملف التنفيذي للوحدة	Clos
	end;
unit About;	<p< td=""></p<>
•	TFmC
interface	TObje
	begin
uses Windows, SysUtils, Classes,	DM1
Graphics, Forms, Controls, StdCtrls,	end;
Buttons, ExtCtrls;	
type	
type TFMAboutBox = class(TForm)	r: TO
Panel1: TPanel;	begin
ProgramIcon: TImage;	DBN
ProductName: TLabel;	DM1.
Version: TLabel;	end;
OKButton: TButton;	
Memo1: TMemo;	<р
<procedure< td=""><td>TFmC</td></procedure<>	TFmC
FormCreate(Sender: TObject);	nder:
private	begin
{ Private declarations }	DBN
public	DM1.
{ Public declarations }	end;
end;	
,	<p< td=""></p<>
var	TFmC
FMAboutBox: TFMAboutBox;	TObj
	begin
implementation	with
	trv

public
{ Public declarations }
end;
var
FmCtrlGrid: TFmCtrlGrid;
implementation
uses DM, About;
{\$R *.dfm}
(4-1-13-13)
<procedure< td=""></procedure<>
TFmCtrlGrid.Button1Click(Sender:
TObject);
begin
Close;
end;
<procedure< td=""></procedure<>
TFmCtrlGrid.FormShow(Sender:
TObject);
begin
DM1.CalculateTotals(Sender, nil);
` ' '
end;
_
<procedure< td=""></procedure<>
TFmCtrlGrid.DBGrid1Enter(Sende
r: TObject);
begin
DBNavigator1.DataSource :=
DM1.DSMaster;
end;
viing
< presedure
<pre><procedure dbctrlcwid1entor(so.)<="" pre="" temctrlcwid=""></procedure></pre>
TFmCtrlGrid.DBCtrlGrid1Enter(Se
nder: TObject);
begin
DBNavigator1.DataSource :=
DM1.DSHoldings;
end;
<procedure< td=""></procedure<>
TFmCtrlGrid.About1Click(Sender:
TObject);
• / /
begin
with TFMAboutBox.Create(nil) do
try

tblMasterINDUSTRY:
TSmallintField;
tblMasterPRICE_CHG:
TSmallintField;
tblMasterRATING: TStringField;
tblMasterRANK: TFloatField;
tblMasterOUTLOOK:
TSmallintField;
tblMasterRCMNDATION:
TStringField;
tblMasterRISK: TStringField;
dsMaster: TDataSource;
tblIndustry: TTable;
tblIndustryIND_CODE:
TSmallintField;
tblIndustryIND_NAME:
TStringField;
tblIndustryLONG_NAME:
TStringField;
dsIndustry: TDataSource;
tblHoldings: TTable;
tblHoldingsACCT NBR:
TFloatField;
tblHoldingsSHARES:
TFloatField;
tblHoldingsPUR PRICE:
TFloatField;
tblHoldingsPUR_DATE:
TDateField;
tblHoldingsSYMBOL:
TStringField;
tblHoldingsPUR COST:
TCurrencyField;
dsHoldings: TDataSource;
<procedure< td=""></procedure<>
tblHoldingsCalcFields(DataSet:
TDataSet);
<procedure< td=""></procedure<>
tblHoldingsAfterPost(DataSet:
TDataSet);
<procedure< td=""></procedure<>
CalculateTotals(Sender: TObject;
Field: TField);
* '
<pre><procedure pre="" thlholdingsafteropen(dataset)<=""></procedure></pre>
tblHoldingsAfterOpen(DataSet:
TDataSet);
<procedure< td=""></procedure<>
DataModuleCreate(Sender:
TObject);

{\$R *.dfm}
<procedure< td=""></procedure<>
TFMAboutBox.FormCreate(Sender:
TObject);
begin
end;
end.
dmالملف التنفيذي للوحدة
-
unit DM;
,
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
DBTables, DB;
type
TDM1 = class(TDataModule)
tblMaster: TTable;
tblMasterSYMBOL:
TStringField;
tblMasterCO NAME:
TStringField;
tblMasterEXCHANGE:
TStringField;
tblMasterIndustryLongName:
TStringField;
tblMasterCUR PRICE:
TFloatField;
tblMasterYRL HIGH:
TFloatField;
tblMasterYRL LOW:
TFloatField;
tblMasterP_E_RATIO:
TFloatField;
tblMasterPROJ GRTH:
TFloatField;
,

displays; otherwise,
clear the result displays. }
if tblHoldings.recordCount = 0 then
begin
{ Clear the result displays }
FmCtrlGrid.lTotalCost.Caption
:= '';
FmCtrlGrid.lTotalShares.Caption
:= '';
FmCtrlGrid.lDifference.Caption
:= ";
end
else
begin
{ let the user know something's
going on }
Screen.Cursor := crHourglass;
9
{ Initialize the holder variables }
flTotalCost := 0.0;
flTotalShares := 0.0;
11100012101101
{ Calculate the total cost of these
holdings. }
tblHoldings.disableControls; {
hide this process from the user }
tblHoldings.first;
while not tblHoldings.eof do
begin
flTotalCost := flTotalCost +
tblHoldingsPUR_COST.AsFloat;
flTotalShares := flTotalShares +
tblHoldingsSHARES.AsFloat;
tblHoldings.next;
-
end;
tblHoldings.first;
tblHoldings.enableControls; {
restore the display of holdings }
{ Calculate the current value of
the shares (by multiplying
the current holdings by the
current share price) and the
difference between the cost and
the value. }
flTotalValue := flTotalShares *
tblMasterCUR PRICE.AsFloat;
flDifference := flTotalValue -

private
{ Private declarations }
public
{ Public declarations }
end;
var
DM1: TDM1;
, and the second
implementation
•
uses CtrlForm;
10000
{\$R *.dfm}
(411 14111)
<procedure< td=""></procedure<>
TDM1.tblHoldingsCalcFields(DataS
et: TDataSet);
begin
tblHoldingsPUR_COST.AsFloat :=
tblHoldingsPUR PRICE.AsFloat
* tblHoldingsSHARES.AsFloat;
end;
enu;
< pre>procedure
<pre><procedure pre="" tdm1.calculatetotals(sender:<=""></procedure></pre>
TObject; Field: TField);
· ·
flTotalCost, { Holds total
,
share cost }
flTotalShares, { Holds total
share count }
flTotalValue, { Holds total
share value }
flDifference: Real; { Holds
difference between cost and value }
strFormatSpec: string; { The
Display Format specification }
begin
{ Update the count of stock
transactions }
FmCtrlGrid.lPurchase.Caption :=
IntToStr(tblHoldings.RecordCount
);
{ See whether or not its necessary
to total the holdings and
(if so) do so and update the result

TDM1.tblHoldingsAfterPost(DataSe	flTotalCost;
t: TDataSet);	
var	{ Use the same format
bmCurrent: TBookmark; { Holds	specification as that being used to
the current position }	display the Current Price field
begin	value so it can be used
with tblHoldings do	to display the results }
begin	
bmCurrent := getBookmark; {	strFormatSpec :=
save position }	tblMasterCUR_PRICE.DisplayFor
try	mat;
CalculateTotals(nil, nil); { recalc	
totals }	{ Update the result displays }
<pre>gotoBookmark(bmCurrent); {</pre>	
restore position }	FmCtrlGrid.lTotalCost.Caption
finally;	:=
freeBookmark(bmCurrent); {	FormatFloat( strFormatSpec,
free memory }	flTotalCost );
end;	FmCtrlGrid.lTotalShares.Caption
end;	:=
end;	FormatFloat( strFormatSpec,
	flTotalValue);
<procedure< td=""><td>FmCtrlGrid.lDifference.Caption</td></procedure<>	FmCtrlGrid.lDifference.Caption
TDM1.tblHoldingsAfterOpen(DataS	:=
et: TDataSet);	FormatFloat( strFormatSpec,
begin	flDifference);
{Don't want this calculation to	
occur until both master & detail are	{ Update the Font Color of the
open}	Diference to
dsMaster.OnDataChange :=	indicate the quality of the
CalculateTotals;	investment }
end;	if flDifference > 0 then
<procedure< td=""><td>FmCtrlGrid.lDifference.Font.Color</td></procedure<>	FmCtrlGrid.lDifference.Font.Color
TDM1.DataModuleCreate(Sender:	:= clGreen
TObject);	else
begin	
	FmCtrlGrid.lDifference.Font.Color
end;	:= clRed;
	FmCtrlGrid.lDifference.update;
end.	
	{ let the user know that we're
111111111111111111111111111111111111111	finished }
	Screen.Cursor := crDefault;
	end;
	end;
. 1. 1	
الملف النصي لبرنامج	

<----procedure

Object

<>object Label6: TLabel
Left = 230
Top = 152
Width = 54
Height = 13
AutoSize = False
Caption = '&Holdings:'
FocusControl = DBEdit2
end
<>object Label7: TLabel
Left = 4
Top = 196
Width = 103
Height = 13
AutoSize = False
Caption = 'Investment Value:'
end
<>object Label8: TLabel
Left = 12
Top = 220
Width = 105
Height = 13
Alignment = taRightJustify
Caption = 'Number of Purchases:'
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -12
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
<>object Label9: TLabel
Left = 15
Top = 242
Width = 82
Height = 13
Alignment = taRightJustify
Caption = 'Total Share Cost:'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -12
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
<>object Label10: TLabel

(CtrlGri)	
إعداد	
علاء الدين اللباد	
واتف ۷۳۷۷۵۶۹۰	

الملف النصي لواجهة البرنامجCtrlGrid
s abis at Em Ctul Cuid.
<>object FmCtrlGrid:
TFmCtrlGrid
Left = 221
Top = 170
ActiveControl = DBGrid1
BorderStyle = bsSingle
Caption = 'Stock Browser'
ClientHeight = 340
ClientWidth = 586
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
Menu = MainMenu1
OldCreateOrder = True
OnCreate = FormCreate
OnShow = FormShow
PixelsPerInch = 96
TextHeight = 13
<>object Bevel1: TBevel
Left = 4
Top = 212
Width = 220
Height = 85
Shape = bsFrame
end
<>object Label5: TLabel
Left = 6
Top = 4
Width = 61
Height = 13
AutoSize = False
Caption = '&Securities:'
FocusControl = DBGrid1
end

Width = 110
Height = 13
Alignment = taRightJustify
AutoSize = False
end
<>object lDifference: TLabel
Left = 103
Top = 279
Width = 110
Height = 13
Alignment = taRightJustify
AutoSize = False
end
<>object DBCtrlGrid1:
TDBCtrlGrid
Left = 228
Top = 168
Width = 351
Height = 159
DataSource = DM1.dsHoldings
PanelHeight = 53
PanelWidth = 334
TabOrder = 1
OnEnter = DBCtrlGrid1Enter
<>object Label1: TLabel
Left = 8
Top = 9
Width = 74
Height = 13
Alignment = taRightJustify
Caption = 'Purchase Date:'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -12
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
<>object Label2: TLabel
Left = 183
Top = 32
Width = 75
Height = 13
Alignment = taRightJustify
Caption = 'Purchase Price:'
Font.Charset =
DEFAULT_CHARSET

Left = 19
Top = 260
Width = 78
Height = 13
Alignment = taRightJustify
Caption = 'Value of Shares:'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -12
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
<>object Label11: TLabel
Left = 23
Top = 279
Width = 74
Height = 13
Alignment = taRightJustify
Caption = 'C/V Difference:'
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -12
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
<>object lPurchase: TLabel
Left = 123
Top = 220
Width = 86
Height = 13
AutoSize = False
end
<>object lTotalCost: TLabel
Left = 103
Top = 242
Width = 110
Height = 13
Alignment = taRightJustify
AutoSize = False
end
<>object lTotalShares:
TLabel
Left = 103
Top = 260
10h =00

W: J4L = 00
Width = 90
Height = 21
DataField = 'PUR_DATE'
DataSource = DM1.dsHoldings
TabOrder = 1
end
<>object DBEdit3: TDBEdit
Left = 259
Top = 29
Width = 66
Height = 21
DataField = 'PUR PRICE'
DataSource = DM1.dsHoldings
TabOrder = 2
end
<>object DBEdit4: TDBEdit
Left = 259
Top = 6
Width = 66
Height = 21
DataField = 'SHARES'
DataSource = DM1.dsHoldings
TabOrder = 3
end
end
<>object DBGrid1: TDBGrid
Left = 4
Top = 20
Width = 573
Height = 128
DataSource = DM1.dsMaster
Options = [dgTitles, dgIndicator,
dgColumnResize, dgColLines,
dgRowLines, dgTabs, dgRowSelect,
dgAlwaysShowSelection,
dgConfirmDelete, dgCancelOnExit]
TabOrder = 0
TitleFont.Charset =
DEFAULT CHARSET
TitleFont.Color = clWindowText
TitleFont.Height = -11 TitleFont Name = 'MS Sons Sovie'
TitleFont.Name = 'MS Sans Serif'
TitleFont.Style = [fsBold]
OnEnter = DBGrid1Enter
Columns = <
item
Expanded = False
FieldName = 'SYMBOL'

Font.Color = clWindowText
Font.Height = -12
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
<>object Label3: TLabel
ŭ
Left = 185
Top = 9
Width = 73
Height = 13
Alignment = taRightJustify
Caption = 'Shares Bought:'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -12
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
<>object Label4: TLabel
Left = 17
Top = 32
Width = 65
Height = 13
Alignment = taRightJustify
Caption = 'Acct Number:'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -12
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
<>object DBEdit1: TDBEdit
9
Left = 83
Top = 29
Width = 90
Height = 21
DataField = 'ACCT_NBR'
DataSource = DM1.dsHoldings
TabOrder = 0
end
<>object DBEdit2: TDBEdit
Left = 83
Top = 6

Width = 64
Visible = True
end
item
Expanded = False
FieldName = 'PRICE CHG'
Width = 64
Visible = True
end
item
Expanded = False
FieldName = 'RATING'
Width = 64
Visible = True
end
V == 4/2
item
Expanded = False
FieldName = 'RANK'
Width = 64
Visible = True
end
item
Expanded = False
FieldName = 'OUTLOOK'
Width = 64
Visible = True
end
item
Expanded = False
FieldName = 'RCMNDATION'
Width = 64
Visible = True
end
item
Expanded = False
FieldName = 'RISK'
Visible = True
end>
end
<>object Button1: TButton
Left = 82
Top = 309
Width = 75
Height = 24
Caption = '&Close'
TabOrder = 2
OnClick = Button1Click
end

Visible = True
end
item
Expanded = False
FieldName = 'CO_NAME'
Width = 64
Visible = True
end
item
Expanded = False
FieldName =
'IndustryLongName'
Width = 64
Visible = True
end
item
Expanded = False
FieldName = 'EXCHANGE'
Width = 64
Visible = True
end
item
Expanded = False
FieldName = 'CUR_PRICE'
Width = 64
Visible = True
end
item
Expanded = False
FieldName = 'YRL_HIGH'
Width = 64
Visible = True
end
item
Expanded = False
FieldName = 'YRL LOW'
Width = 64
Visible = True
end
item
Expanded = False
FieldName = 'P E RATIO'
Width = 64
Visible = True
end
item
Expanded = False
FieldName = 'PROJ_GRTH'

Width = 281
Height = 265
BevelInner = bvRaised
BevelOuter = bvLowered
TabOrder = 0
<>object ProgramIcon:
TImage Left = 8
Top = 8 Width = 65
Width = 65 Height = 57
Picture.Data =000
0000}
Stretch = True
IsControl = True
end
<>object ProductName:
TLabel
Left = 88
Top = 16
Width = 92
Height = 13
Caption = 'DBCtrlGrid Example'
IsControl = True
end
<>object Version: TLabel
Left = 88
Top = 40
Width = 53
Height = 13
Caption = 'Version 1.0'
IsControl = True
end
<>object Memo1: TMemo
Left = 8
Top = 72
Width = 265
Height = 181
Lines.Strings = (
'This example illustrates'
"
'- The use of the DBCtrlGrid'
'- Calculating and Displaying
Total Fields'
'- Formatting of fields using
the format spec.'
'- Using Bookmarks'
'- Defining an Event Handler

<>object DBNavigator1:
TDBNavigator
Left = 4
Top = 168
Width = 210
Height = 18
TabOrder = 3
end
<>object MainMenu1:
TMainMenu
Left = 165
Top = 224
<>object About1:
TMenuItem
Caption = '&About'
OnClick = About1Click
end
end
End
لAboutBox الملف النصى
- TIDOUEDONG
Z Sabiast EM AbautDays
<>object FMAboutBox:
TFMAboutBox
Left = 189
Top = 185
BorderStyle = bsDialog
Caption = 'About'
ClientHeight = 321
ClientWidth = 298
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<>object Panel1: TPanel
Left = 8
Top = 8

tblMasterSYMBOL: TStringField
Alignment = taCenter
DisplayLabel = 'Symbol'
DisplayWidth = 7
FieldName = 'SYMBOL'
Size = 7
end
<>object
tblMasterCO_NAME: TStringField
DisplayLabel = 'Company'
DisplayWidth = 20
FieldName = 'CO NAME'
end
<>object
tblMasterEXCHANGE:
TStringField
DisplayLabel = 'Exchange'
DisplayWidth = 8
FieldName = 'EXCHANGE'
Size = 8
end
<>object
tblMasterIndustryLongName:
TStringField
Alignment = taCenter
DisplayLabel = 'Industry'
DisplayWidth = 18
FieldKind = fkLookup
FieldName =
'IndustryLongName'
LookupDataSet = tblIndustry
LookupKeyFields =
'IND CODE'
LookupResultField =
'LONG NAME'
KeyFields = 'INDUSTRY'
Size = 32
Lookup = True
end
<>object
tblMasterCUR PRICE: TFloatField
DisplayLabel = 'Current Price'
DisplayWidth = 10
FieldName = 'CUR_PRICE'
DisplayFormat = '\$ #,###.00'
EditFormat = '#,###.00'
end
<>object

at runtime'
"
'CtrlGrid uses the Stock,
Holdings and Industry '
'tables from the DBDemos data
to display a '
'listing of portfolio investments
and their current '
'valuation.'
valuation.
D 10 1 T
ReadOnly = True
ScrollBars = ssVertical
TabOrder = 0
end
end
<>object OKButton: TButton
Left = 103
Top = 284
Width = 77
Height = 27
Caption = 'OK'
Default = True
ModalResult = 1
TabOrder = 1
end
End
e, tt
dm1الملف النصي للوحدة
<>object DM1: TDM1
OldCreateOrder = True
OnCreate = DataModuleCreate
Left = 315
Top = 359
Height = 179
Width = 202
<>object tblMaster: TTable
Active = True
DatabaseName = 'DBDEMOS'
IndexName = SYMBOL
IndexName = 'SYMBOL' TableName = 'MASTER.DBF'
TableName = 'MASTER.DBF'
TableName = 'MASTER.DBF' Left = 24
TableName = 'MASTER.DBF'

DisplayFormat = '###%'
EditFormat = '###'
end
<>object
tblMasterRATING: TStringField
Alignment = taCenter
DisplayLabel = 'Rating'
DisplayWidth = 5
FieldName = 'RATING'
Size = 4
end
<>object tblMasterRANK:
TFloatField
Alignment = taCenter
DisplayLabel = 'Rank'
DisplayWidth = 5
FieldName = 'RANK'
end
<>object
tblMasterOUTLOOK:
TSmallintField
Alignment = taCenter
DisplayLabel = 'Outlook'
DisplayWidth = 7
FieldName = 'OUTLOOK'
end
<>object
tblMasterRCMNDATION:
TStringField
Alignment = taCenter
DisplayLabel = 'Recc.'
DisplayWidth = 6
FieldName = 'RCMNDATION'
Size = 5
end DISW A DISW
<>object tblMasterRISK:
TStringField
Alignment = taCenter
DisplayLabel = 'Risk'
DisplayWidth = 6
FieldName = 'RISK'
Size = 4
end
end
<>object dsMaster:
TDataSource
DataSet = tblMaster
Left = 76

tblMasterYRL HIGH: TFloatField
DisplayLabel = 'Yr. High'
DisplayWidth = 10
FieldName = 'YRL_HIGH'
DisplayFormat = '\$ #,###.00'
EditFormat = '#,###.00'
end
<>object
tblMasterYRL LOW: TFloatField
DisplayLabel = 'Yr. Low'
DisplayWidth = 10
FieldName = 'YRL_LOW'
DisplayFormat = '\$ #,###.00'
EditFormat = '#,###.00'
end
<>object
tblMasterP_E_RATIO: TFloatField
Alignment = taCenter
DisplayLabel = 'P/E Ratio'
DisplayWidth = 10
FieldName = 'P_E_RATIO'
end
<>object
tblMasterPROJ_GRTH:
TFloatField
Alignment = taCenter
DisplayLabel = 'Proj. Growth'
DisplayWidth = 10
FieldName = 'PROJ_GRTH'
DisplayFormat = '###%'
EditFormat = '###'
end
<>object
tblMasterINDUSTRY:
TSmallintField
Alignment = taCenter
DisplayLabel = 'Industry'
DisplayWidth = 10
FieldName = 'INDUSTRY'
Visible = False
end
<>object
tblMasterPRICE CHG:
TSmallintField
Alignment = taCenter
DisplayLabel = 'Price Change'
DisplayWidth = 10
FieldName = 'PRICE_CHG'

<>object
tblHoldingsACCT_NBR:
TFloatField
DisplayLabel = 'Account No.'
FieldName = 'ACCT_NBR'
end
<>object
tblHoldingsSHARES: TFloatField
DisplayLabel = 'Shares'
FieldName = 'SHARES'
DisplayFormat = '###,###,###'
EditFormat = '###,###,###'
end
<>object
tblHoldingsPUR_PRICE:
TFloatField
DisplayLabel = 'Purchase Price'
FieldName = 'PUR_PRICE'
DisplayFormat = '\$ #,###.00'
EditFormat = '#,###.00'
end
<>object
tblHoldingsPUR_DATE: TDateField
DisplayLabel = 'Purchase Date'
FieldName = 'PUR_DATE'
DisplayFormat = 'mmm. dd,
yyyy'
end
<>object
tblHoldingsSYMBOL: TStringField
FieldName = 'SYMBOL'
Visible = False
Size = 7
end
<>object
tblHoldingsPUR_COST:
TCurrencyField
FieldKind = fkCalculated
FieldName = 'PUR_COST'
Calculated = True
end
end
<>object dsHoldings:
<b>TDataSource</b>
DataSet = tblHoldings
Left = 80
Top = 68
end
End

Top = 12
end
<>object tblIndustry: TTable
Active = True
DatabaseName = 'DBDEMOS'
IndexName = 'IND CODE'
MasterFields = 'INDUSTRY'
TableName = 'INDUSTRY.DBF'
Left = 140
Top = 12 <>object
tblIndustryIND_CODE:
TSmallintField TSmallintField
FieldName = 'IND_CODE'
Visible = False
end
<>object
tblIndustryIND_NAME:
TStringField
FieldName = 'IND NAME'
Size = 5
end
<>object
tblIndustryLONG NAME:
TStringField
FieldName = 'LONG NAME'
end
end
<>object dsIndustry:
TDataSource
DataSet = tblIndustry
Left = 136
Top = 68
end
<>object tblHoldings: TTable
Active = True
AfterOpen =
tblHoldingsAfterOpen
AfterPost = tblHoldingsAfterPost
OnCalcFields =
tblHoldingsCalcFields
DatabaseName = 'DBDEMOS'
IndexName = 'SYMBOL'
MasterFields = 'SYMBOL'
MasterSource = dsMaster
TableName = 'HOLDINGS.DBF'
Left = 12
Top = 68

Forms, StdCtrls, DBCtrls,
DBGrids, DB, DBTables, Buttons,
Grids, ExtCtrls;
type
TForm1 = class(TForm)
Panel1: TPanel;
Label1: TLabel;
DBImage1: TDBImage;
DBLabel1: TDBText;
DBMemo1: TDBMemo;
DataSource1: TDataSource;
Table1: TTable;
Table1Common Name:
TStringField;
Table1Graphic: TBlobField;
DBGrid1: TDBGrid;
BitBtn1: TBitBtn;
Table1Category: TStringField;
Table1SpeciesName:
TStringField;
Table1Lengthcm: TFloatField;
Table1Length_In: TFloatField;
Table1Notes: TMemoField;
procedure FormCreate(Sender:
TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form1: TForm1;
,
implementation
•
{\$R *.dfm}
procedure
TForm1.FormCreate(Sender:
TObject);
begin
end;
end.

انتهى البرنامج والآن إلى الصور المساعدة

4FishFact الملف التنفيذي للبرنامج:
FishFact
unit Ffactwin;
{ This application shows how to
display Paradox style memo and
graphic
fields in a form. Table1's
DatabaseName property should
point to the
Delphi sample database. Table1's
TableName property should be set to
the BIOLIFE table. }
interface
uses
SysUtils, Windows, Messages,
Classes, Graphics, Controls,

T C C
Left = 6
Top = 8
Width = 259
Height = 217
Hint = 'Scroll grid below to see
other fish'
ParentShowHint = False
ShowHint = True
TabOrder = 0
<object dblabel1:<="" td=""></object>
TDBText
Left = 4
Top = 183
Width = 249
Height = 24
DataField = 'Common Name'
DataSource = DataSource1
Font.Charset =
DEFAULT CHARSET
Font.Color = clRed
Font.Height = -19
Font.Name = 'MS Serif'
Font.Style = [fsBold, fsItalic]
ParentFont = False
end
<object dbimage1:<="" td=""></object>
TDBImage Left = 5
Top = 8
Width = 248
Height = 168
Hint = 'Scroll grid below to see
other fish'
DataField = 'Graphic'
DataSource = DataSource1
TabOrder = 0
end
end
object Panel2: TPanel
Left = 275
Top = 8
Width = 223
Height = 22
TabOrder = 1
<object label1:="" td="" tlabel<=""></object>
Left = 7
Top = 4
Width = 56
William 50

#### &FishFact

#### 4FishFact

4FishFact
program Fishfact;
إعداد علاء الدين اللباد
.9 £ £ 0 V 0 T V 1
الملف النصي
Uses
OSCS
Louma
Forms, Ffactwin in 'FFACTWIN.PAS'
{Form1};
{\$R *.RES}
begin
Application.Initialize;
Application.CreateForm(TForm1,
Form1);
Application.Run;
end.
- Cirui
1. 4E 1 E 1
object Form1: TForm1
Left = 223
Top = 158
ActiveControl = DBImage1
BorderIcons = [biSystemMenu,
biMinimize]
BorderStyle = bsSingle
Caption = 'FISH FACTS'
ClientHeight = 394
ClientWidth = 596
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
<del>_</del>
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
ShowHint = True
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
object Panel1: TPanel
~~Joor 1 mileste 1 1 miles

E (C) IDI I
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentCtl3D = False
ParentFont = False
ScrollBars = ssVertical
TabOrder = 0
end
end
object Panel4: TPanel
Left = 0
Top = 322
Width = 596
Height = 72
Align = alBottom
BevelInner = bvRaised
BorderStyle = bsSingle
ParentShowHint = False
ShowHint = True
TabOrder = 3
<object dbgrid1:<="" td=""></object>
TDBGrid
Left = 12
Top = 8 Width = 386
Height = 53
Hint = 'Scroll up/down to see
other fish!'
DataSource = DataSource1
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 0
TitleFont.Charset =
DEFAULT_CHARSET
TitleFont.Color = clBlack
TitleFont.Height = -11
TitleFont.Name = 'MS Sans
Serif'
TitleFont.Style = []
end
<object bitbtn1:="" tbitbtn<="" td=""></object>
Left = 424

Height = 13
Caption = 'About the'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
<object dblabel2:<="" td=""></object>
TDBText
Left = 67
Top = 4
Width = 99
Height = 13
AutoSize = True
DataField = 'Common_Name'
DataSource = DataSource1
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
end
object Panel3: TPanel
Left = 276
Top = 32
Width = 223
Height = 193
BevelOuter = bvLowered
TabOrder = 2
<object dbmemo1:<="" td=""></object>
TDBMemo
Left = 3
Top = 2
Width = 217
Height = 183
BorderStyle = bsNone
Color = clSilver
Ctl3D = False
DataField = 'Notes'
DataSource = DataSource1
Font.Charset =
DEFAULT_CHARSET

الهدف من هذا البرنامج هو تعريف الطالب بكيفية التعامل مع البيانات المخزنة ضمن data base وذلك بإدخالها ضمن برنامج الدلفي من اجل حمايتها أولا ومن ثم إظهارها بالمظهر اللائق وتصميم الواجهة حسب ذوق المستثمر وإظهار الإبداع لدى المبرمج من اجل إيصال الفكرة المقصودة من البرنامج و قد تم شرح هذا البرنامج في خطوة ويجب الانتباه إلى ما يلي في أي قاعدة بيانات مستوردة:

۱- في التعليمة Table1 يجب ان يكون:

object Table1: TTable		
Active = True		
DatabaseName = 'DBDEMOS		
Name=table1'		
ReadOnly = True		
TableName = 'BIOLIFE'		
Left = 221		
Top = 193		

Top = 20
Width = 57
Height = 29
Hint = 'Close Fish Facts'
Caption = 'E&xit'
TabOrder = 1
Kind = bkClose
end
end
object DataSource1: TDataSource
DataSet = Table1
Left = 19
Top = 193
end
object Table1: TTable
Active = True
DatabaseName = 'DBDEMOS'
ReadOnly = True
TableName = 'BIOLIFE'
Left = 221
Top = 193
<object table1category:<="" td=""></object>
TStringField
DisplayWidth = 15
FieldName = 'Category' Size = 15
end
<object< td=""></object<>
Table1SpeciesName: TStringField
DisplayWidth = 20
FieldName = 'Species Name'
Size = 40
end
<object table1lengthcm:<="" td=""></object>
TFloatField
DisplayWidth = 11
FieldName = 'Length (cm)'
end
<object table1length_in:<="" td=""></object>
TFloatField
DisplayWidth = 10
FieldName = 'Length_In'
DisplayFormat = '0.00'
end
<object< td=""></object<>
Table1Common_Name:
TStringField
DisplayWidth = 30

٤- يجب ان يتم التعامل مع التعليمات السابقة حسب التسلسل أي نبدأ من الي ٣ وبالنسبة للأزرار الموجودة على الشكل فهي موزعة على الصفحات كما يلى:

object Form1: TForm1:

standard

DBLabel1: datacontrol

dataAccess,Bde,Additional أي ان التعامل مع ٥ صفحات فقط

ثم ننقر بزر الفارة الأيمن على التعليمة table1 ثم نختار الحقول المطلوبة ونقر add

# ٢- في التعليمة DataSource1يجب ان يكون :

object DataSource1: TDataSource			
DataSet = Table1			
Tablename=biolife.db			
Active=true			
Name = DataSource1			
Databasename=DBDemos			
Left = 19			
Top = 193			
end			

## ٣- في التعليمة Dbgrid1 يجب أنيكون:

· • • • • • • • • • • • • • • • • • •
object DBGrid1: TDBGrid
Left = 12
Top = 8
Width = 386
Height = 53
Hint = 'Scroll up/down to see
other fish!'
DataSource = DataSource1
Dataset = table1
Active = true
Databasename=DBDemos
Tablename=biolife.db
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 0
TitleFont.Charset =
DEFAULT_CHARSET
TitleFont.Color = clBlack
TitleFont.Height = -11
TitleFont.Name = 'MS Sans
Serif'
TitleFont.Style = []
end

unit bmpformu;
// A TBitmap is an encapsulation of
the windows BITMAP and
PALETTE which
// manages realizing of the palette
automatically.
//
// The bitmap can be loaded from a
.BMP file (LoadFromFile method)
or a
// resource
(LoadFromResourceName or
LoadFromResourceID method) and
saved
// back to a file (SaveToFile method).
It can be drawn on a canvas by
// using the TCanvas' Draw or
StretchDraw method. The size of
the bitmap
// can be determined by using the
Height and Width properties of
TBitmap.
//
// The example below illustrates the
use of the following:
//
// TBitmap, Palette, LoadFromFile,
Draw, Height, Width
// Do supe that a small hitman file
// Be sure that a small bitmap file
named bor6.bmp is present in the
same
// directory as the .exe file.
// Disclaimer: This is an example of
<u>-</u>
using TBitmaps and is not intended
to
// be an efficient method of tiling a
form.
interface
111111111111
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
StdCtrls;
State 1159

ObjectS			
نفيذي	الهلف الكاهل النصي والت		
برنامج	1		
(6Bitn	nap)		
إعداد			
علاء الدين اللباد			
واتف ۷۳۷۱۹۹۰			
ون الخلفية	برنامج لتغيير ل		
1			
التنفيذي			
Procedu	res 12		

	A	
// since we're going to be painting the	type	
whole form, handling this	TBmpForm = class(TForm)	
// message will suppress the	Button1: TButton;	
uneccessary repainting of the	<> procedure	
background	FormDestroy(Sender: TObject);	
// which can result in flicker.	<> procedure	
<> procedure	FormPaint(Sender: TObject);	
TBmpform.WMEraseBkgnd(var m :	<> procedure	
TWMEraseBkgnd);	Button1Click(Sender: TObject);	
begin	<> procedure	
m.Result := LRESULT(False);	FormCreate(Sender: TObject);	
end;	private	
	{ Private declarations }	
<> procedure	Bitmap: TBitmap;	
TBmpForm.FormPaint(Sender:	<> procedure	
TObject);	ScrambleBitmap;	
var	<> procedure	
x, y: Integer;	WMEraseBkgnd(var m:	
begin	TWMEraseBkgnd); message	
y := 0;	WM ERASEBKGND;	
while y < Height do	Public	
begin	{ Public declarations }	
x := 0;	end;	
while x < Width do	Cira,	
begin	Var	
Canvas.Draw(x, y, Bitmap);	BmpForm: TBmpForm;	
x := x + Bitmap.Width;	Disprorm. 1Disprorm,	
end;	Implementation	
y := y + Bitmap.Height;	Implementation	
end;	(CD * DEM)	
end;	{\$R *.DFM}	
<> procedure	Z S nuccedure	
TBmpForm.Button1Click(Sender:	<> procedure	
TObject);	TBmpForm.FormCreate(Sender:	
begin	TObject);	
ScrambleBitmap;	begin This Control	
Invalidate;	Bitmap := TBitmap.Create;	
end;		
	Bitmap.LoadFromFile('bor6.bmp');	
	end;	
//		
// scrambling the bitmap is easy	<> procedure	
when it's has 256 colors:	TBmpForm.FormDestroy(Sender:	
// we just need to change each of the	TObject);	
color in the palette	begin	
// to some other value.	Bitmap.Free;	
//	end;	
<> procedure		
	<del></del>	

Font.Style = []
OldCreateOrder = True
OnCreate = FormCreate
OnDestroy = FormDestroy
OnPaint = FormPaint
PixelsPerInch = 96
TextHeight = 13
object Button1: TButton
Left = 56
Top = 40
Width = 113
Height = 25
Caption = 'Scramble Palette'
TabOrder = 0
OnClick = Button1Click
end
End

TBmpForm.ScrambleBitmap;
var
pal: PLogPalette;
hpal: HPALETTE;
i: Integer;
begin
pal := nil;
try
GetMem(pal, sizeof(TLogPalette)
+ sizeof(TPaletteEntry) * 255);
pal.palVersion := \$300;
pal.palNumEntries := 256;
for $i := 0$ to 255 do
begin
pal.palPalEntry[i].peRed :=
Random(255);
pal.palPalEntry[i].peGreen :=
Random(255);
pal.palPalEntry[i].peBlue :=
Random(255);
end;
hpal := CreatePalette(pal^);
if hpal $\Leftrightarrow$ 0 then
Bitmap.Palette := hpal;
finally
FreeMem(pal);
end;
end;
end.
الملف النصي للبرنامج
object BmpForm: TBmpForm
Left = 136
Top = 124
Width = 466
Height = 334
Caption = 'BmpForm'
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, ComCtrls, Menus, ExtCtrls;  type TComboForm = class(TForm)
Dialogs, StdCtrls, ComCtrls, Menus, ExtCtrls;  type TComboForm = class(TForm)
StdCtrls, ComCtrls, Menus, ExtCtrls;  type TComboForm = class(TForm)
type TComboForm = class(TForm)
type TComboForm = class(TForm)
TComboForm = class(TForm)
TComboForm = class(TForm)
CbDrop: TComboBox;
CbDropList: TComboBox;
CbSimple: TComboBox;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
MainMenu1: TMainMenu;
ComboStyle: TMenuItem;
CmSimple: TMenuItem;
CmDrop: TMenuItem;
CmDropList: TMenuItem;
Add1: TMenuItem;
CmAddString: TMenuItem;
CmAddStringAt: TMenuItem;
Search1: TMenuItem;
CmFindString: TMenuItem;
CmFindIndex: TMenuItem;
CmDelete: TMenuItem;
CmList: TMenuItem;
CmDelString: TMenuItem;
CmDelIndex: TMenuItem;
CmShowList: TMenuItem;
CmClear: TMenuItem;
N1: TMenuItem;
Label5: TLabel;
EdCursel: TEdit;
EdCuridx: TEdit;
EdCurlen: TEdit;
EdEdit: TEdit;
EdEditlen: TEdit;
CmSortList: TMenuItem;
Bevel1: TBevel;
procedure
ComboStyleClick(Sender: TObject);
procedure
CmSimpleClick(Sender: TObject);
procedure CmDropClick(Sender:
TObject);
procedure

<b>Object</b> §	
الهلف التنفيذي لبرناهج	
(Combobox)	
إعداد	
علاء الدين اللباد	
واتف ۷۳۷۱۹۹۱	
unit combform;	
interface	
uses	

nuan auty Ch A ativas TC amb a Day
property CbActive: TComboBox
read FCbActive write SetCbActive;
public
{ Public declarations }
end;
Wan
Comba Francis TC and a Francis
ComboForm: TComboForm;
implementation
uses inptform;
_
{\$R *.DFM}
(421 (221)2)
////
// event handlers //
////
procedure
TComboForm.FormCreate(Sender:
TObject);
begin
FCbActive := CbSimple;
CbSimple.Visible := True;
CbDrop.Visible := False;
CbDropList.Visible := False;
NewSelection;
end;
// Control notifications
//
//
// close form
//
procedure
TComboForm.Button1Click(Sender:
TObject);
begin
Close;
end;
Ciiu,
procedure
TComboForm.ComboChange(Sende
r: TObject);
begin

CmDropListClick(Sender:
TObject);
procedure FormCreate(Sender:
TObject);
procedure ComboChange(Sender:
TObject);
procedure CmClearClick(Sender:
TObject);
procedure
CmDelStringClick(Sender:
TObject);
procedure
CmDelIndexClick(Sender:
TObject);
<b>4</b>
procedure
CmFindStringClick(Sender:
TObject);
procedure
CmFindIndexClick(Sender:
TObject);
procedure
CmAddStringClick(Sender:
TObject);
procedure
CmAddStringAtClick(Sender:
TObject);
procedure Button1Click(Sender:
TObject);
procedure
CmSortListClick(Sender: TObject);
procedure CmListClick(Sender:
TObject);
procedure
CmShowListClick(Sender:
TObject);
private
{ Private declarations }
FCbActive: TComboBox;
procedure SetCbActive(cb:
TComboBox);
procedure NewSelection;
procedure DeleteAt(idx: Integer);
procedure AddAt(idx: Integer);
string);
G/ *
procedure SelectAt(idx: Integer);
e To 100
function FindString(s: string):
Integer;

add:', s) then	NewSelection;
AddAt(0, s);	end;
end;	
procedure	// ComboBox menu
TComboForm.CmAddStringAtClick	//
(Sender: TObject);	
var	procedure
i: Integer;	TComboForm.ComboStyleClick(Sen
s: string;	der: TObject);
begin	begin
if InputForm.GetString('String to	CmSimple.Checked := CbActive =
add:', s)	CbSimple;
and	CmDrop.Checked := CbActive =
InputForm.GetInteger('Index to	CbDrop;
insert at:', i) then	CmDropList.Checked := CbActive
AddAt(i, s);	= CbDropList;
end;	end;
	,
// Find menu	procedure
//	TComboForm.CmSimpleClick(Send
procedure	er: TObject);
TComboForm.CmFindStringClick(S	begin
ender: TObject);	CbActive := CbSimple;
var	end;
s: string;	
begin	procedure
if InputForm.GetString('String to	TComboForm.CmDropClick(Sender
select:', s) then	: TObject);
SelectAt(FindString(s));	begin
end;	CbActive := CbDrop;
	end;
procedure	,
TComboForm.CmFindIndexClick(S	procedure
ender: TObject);	TComboForm.CmDropListClick(Se
var	nder: TObject);
i: Integer;	begin
begin	CbActive := CbDropList;
if InputForm.GetInteger('Index to	end;
select:', i) then	
SelectAt(i);	// Add menu
end;	//
	procedure
	TComboForm.CmAddStringClick(S
// Delete menu	ender: TObject);
// Detect menu	var
procedure	s: string;
TComboForm.CmDelStringClick(Se	begin
nder: TObject);	if InputForm.GetString('String to
	Crime of the mile to

CmShowList.Enabled := CbActive	var
<> CbSimple;	s: string;
CmSortList.Checked :=	begin
CbActive.Sorted;	if InputForm.GetString('String to
end;	delete:', s) then
	DeleteAt(FindString(s));
	end;
////	,
// private methods //	procedure
////	TComboForm.CmDelIndexClick(Se
	nder: TObject);
procedure	var
TComboForm.NewSelection;	i: Integer;
begin	begin
if CbActive.ItemIndex = -1 then	if InputForm.GetInteger('Index to
EdCursel.Text := "	delete:', i) then
else	DeleteAt(i);
EdCursel.Text :=	end;
CbActive.Items[CbActive.ItemIndex	
];	procedure
EdCuridx.Text :=	TComboForm.CmClearClick(Sende
IntToStr(CbActive.ItemIndex);	r: TObject);
EdCurlen.Text :=	begin
IntToStr(length(EdCursel.Text));	CbActive.Clear;
	end;
EdEdit.Text := CbActive.Text;	,
EdEditlen.Text :=	
IntToStr(length(EdEdit.Text));	// List menu
end;	//
,	procedure
procedure	TComboForm.CmShowListClick(Se
TComboForm.SetCbActive(cb:	nder: TObject);
TComboBox);	begin
begin	CbActive.DroppedDown := not
if FCbActive <> cb then	CbActive.DroppedDown;
begin	end;
FCbActive.Visible := False;	,
FCbActive := cb;	procedure
FCbActive.Visible := True;	TComboForm.CmSortListClick(Sen
NewSelection;	der: TObject);
end;	begin
end;	CbActive.Sorted := not
	CbActive.Sorted;
	end;
//	,
// Delete an item from the active	procedure
combobox given its index.	TComboForm.CmListClick(Sender:
// Ignores bogus indices.	TObject);
// <b></b>	begin
·	

1.
//
function
TComboForm.FindString(s: string):
Integer;
var
i: Integer;
begin
i := 0;
while (i < CbActive.Items.Count)
and (CbActive.Items[i] ⇔ s) do
inc(i);
Result := i;
end;
end.

ObjectS
الهلف النصي لبرناهج
(Combobox)
إعداد
علاء الدين اللباد
واتف ۷۳۷۱۹۹۱
object ComboForm: TComboForm
Left = 227
Top = 141
BorderIcons = [biSystemMenu,
biMinimize]
BorderStyle = bsSingle
Caption = 'ComboBox demo'
ClientHeight = 281
ClientWidth = 611
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
Menu = MainMenu1

procedure
TComboForm.DeleteAt(idx:
Integer);
begin
if $idx \ge 0$ then
CbActive.Items.Delete(idx);
` ' '
end;
//
// Add a string into the active
combobox at the given index.
// If the index is too big, append the
item, if too small,
// insert it at the top.
//
procedure
TComboForm.AddAt(idx: Integer;
s: string);
begin
if idx > CbActive.Items.Count then
CbActive.Items.Append(s)
else
begin
if idx < 0 then
idx := 0;
CbActive.Items.Insert(idx, s);
end;
end;
//
// Select an item in the active
combobox given it's index.
// A bogus index will result in an
exception.
procedure
TComboForm.SelectAt(idx:
Integer);
begin
CbActive.ItemIndex := idx;
NewSelection;
end;
Ciru,
11
//
// Find the index of an item in the
combobox given a string.
// If the string does not exist, return -

Left = 40
Top = 48
Width = 178
Height = 21
ItemHeight = 13
TabOrder = 0
Visible = False
OnChange = ComboChange
Items.Strings = (
'in' 'this'
'box'
'we'
'have'
'a'
a 'list'
'of'
'strings') end
object CbDropList: TComboBox
Left = 40
Top = 48
Width = 178
Height = 21
Style = csDropDownList
ItemHeight = 13
TabOrder = 1
Visible = False
OnChange = ComboChange
Items.Strings = (
'Closer'
'Still'
'Substance'
'Little Earthquakes'
'Under the Pink'
'Boys for Pele')
end
object CbSimple: TComboBox
Left = 40
Top = 48
Width = 178
Height = 129
Style = csSimple
ItemHeight = 13
TabOrder = 2
OnChange = ComboChange
Items.Strings = (
'Return of the Native'

OldCreateOrder = True
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
object Label1: TLabel
Left = 288
Top = 48
Width = 82
Height = 13
Caption = 'Current selection:'
end
object Label2: TLabel
Left = 305
Top = 82
Width = 65
Height = 13
Caption = 'Current index:'
end
object Label3: TLabel
Left = 241
Top = 116 Width = 129
Height = 13
Caption = 'Length of current selection:'
end
object Label4: TLabel
Left = 321
Top = 150 Width = 49
W 10th - 49
$H_{a}$ : $abA = 12$
Height = 13
Caption = 'Edit string:'
Caption = 'Edit string:'
Caption = 'Edit string:' end object Label5: TLabel
Caption = 'Edit string:' end object Label5: TLabel Left = 271
Caption = 'Edit string:' end object Label5: TLabel Left = 271 Top = 176
Caption = 'Edit string:' end object Label5: TLabel Left = 271 Top = 176 Width = 99
Caption = 'Edit string:' end object Label5: TLabel Left = 271 Top = 176 Width = 99 Height = 13
Caption = 'Edit string:' end object Label5: TLabel Left = 271 Top = 176 Width = 99 Height = 13 Caption = 'Length of Edit String:'
Caption = 'Edit string:' end object Label5: TLabel Left = 271 Top = 176 Width = 99 Height = 13 Caption = 'Length of Edit String:' end
Caption = 'Edit string:' end object Label5: TLabel Left = 271 Top = 176 Width = 99 Height = 13 Caption = 'Length of Edit String:' end object Bevel1: TBevel
Caption = 'Edit string:' end object Label5: TLabel Left = 271 Top = 176 Width = 99 Height = 13 Caption = 'Length of Edit String:' end object Bevel1: TBevel Left = 0
Caption = 'Edit string:' end object Label5: TLabel Left = 271 Top = 176 Width = 99 Height = 13 Caption = 'Length of Edit String:' end object Bevel1: TBevel Left = 0 Top = 0
Caption = 'Edit string:' end object Label5: TLabel Left = 271 Top = 176 Width = 99 Height = 13 Caption = 'Length of Edit String:' end object Bevel1: TBevel Left = 0 Top = 0 Width = 611
Caption = 'Edit string:' end object Label5: TLabel Left = 271 Top = 176 Width = 99 Height = 13 Caption = 'Length of Edit String:' end object Bevel1: TBevel Left = 0 Top = 0 Width = 611 Height = 2
Caption = 'Edit string:' end object Label5: TLabel Left = 271 Top = 176 Width = 99 Height = 13 Caption = 'Length of Edit String:' end object Bevel1: TBevel Left = 0 Top = 0 Width = 611
Caption = 'Edit string:' end object Label5: TLabel Left = 271 Top = 176 Width = 99 Height = 13 Caption = 'Length of Edit String:' end object Bevel1: TBevel Left = 0 Top = 0 Width = 611 Height = 2

Top = 8
object ComboStyle: TMenuItem
Caption = '&ComboBox'
OnClick = ComboStyleClick
object CmSimple: TMenuItem
Caption = '&Simple'
OnClick = CmSimpleClick
end
object CmDrop: TMenuItem
Caption = '&Dropdown'
OnClick = CmDropClick
end
object CmDropList: TMenuItem
Caption = 'Dropdown &list'
OnClick = CmDropListClick
end
end
object Add1: TMenuItem
Caption = '&Add'
object CmAddString:
TMenuItem
Caption = '&String'
OnClick = CmAddStringClick
end
object CmAddStringAt:
TMenuItem
Caption = 'String &at'
OnClick =
CmAddStringAtClick
end
end
object Search1: TMenuItem
Caption = '&Find'
object CmFindString:
TMenuItem
Caption = '&String'
OnClick = CmFindStringClick
end
object CmFindIndex:
TMenuItem
Caption = '&Index'
OnClick = CmFindIndexClick
end
end
object CmDelete: TMenuItem
Caption = '&Delete'
object CmDelString: TMenuItem
Caption = '&String'

'Jude the Obscure'
'One Hundred Years of Solitude'
'The Autumn of the Patriarch'
'Idoru'
'Neuromancer')
end
object EdCursel: TEdit
Left = 384
Top = 48
Width = 178
Height = 21
ReadOnly = True
TabOrder = 3
end
object EdCuridx: TEdit
Left = 384
Top = 80
Width = 178
Height = 21
ReadOnly = True
TabOrder = 4
end
object EdCurlen: TEdit
Left = 384
Top = 112
Width = 178
Height = 21
ReadOnly = True
TabOrder = 5
end
object EdEdit: TEdit
Left = 384
Top = 144
Width = 178
Height = 21
ReadOnly = True
TabOrder = 6
end
object EdEditlen: TEdit
Left = 384
Top = 176
Width = 178
Height = 21
ReadOnly = True
TabOrder = 7
end
object MainMenu1: TMainMenu
Left = 8

Dil-DI 06
PixelsPerInch = 96
TextHeight = 13
object PromptLabel: TLabel Left = 40
Top = 8
Width = 5
Height = 13
Font.Charset =
DEFAULT_CHARSET Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
object InputEdit: TEdit
Left = 36
Top = 24
Width = 193
Height = 21
TabOrder = 0
end
object Button1: TButton
Left = 43
Top = 72
Width = 75
Height = 25
Caption = '&OK'
Default = True
ModalResult = 1
TabOrder = 1
end
object Button2: TButton
Left = 147
Top = 72
Width = 75
Height = 25
Caption = '&Cancel'
ModalResult = 2
TabOrder = 2
end
End

OnClick = CmDelStringClick
end
object CmDelIndex: TMenuItem
Caption = '&Index'
OnClick = CmDelIndexClick
end
object N1: TMenuItem
Caption = '-'
end
object CmClear: TMenuItem
Caption = '&All'
OnClick = CmClearClick
end
end
object CmList: TMenuItem
Caption = '&List'
OnClick = CmListClick
object CmShowList: TMenuItem
Caption = '&Show'
OnClick = CmShowListClick
end
object CmSortList: TMenuItem
Caption = 'So&rt'
OnClick = CmSortListClick
end
end
end
Cita
End
Litu
111111111111111111111111111111111111111
object InputForm: TInputForm
Left = 150
Top = 147
Width = 274
Height = 140
ActiveControl = InputEdit
Caption = 'InputForm'
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Fant Name - IMC Cans Cariff
Font.Name = 'MS Sans Serif'
Font.Name = Wis Sans Serii Font.Style = []
Font.Style = []

Forms,
Main in 'Main.pas' {FmMain},
DM1 in 'Dm1.pas' {DM},
About in 'About.pas'
{FmAboutBox};
{\$R *.RES}
begin
Application.Initialize;
Application.CreateForm(TFmMain,
FmMain);
Application.CreateForm(TDM,
DM);
Application.Run;
end.
111111111111111111111111111111111111111
unit Main;
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
Grids, DBGrids, ExtCtrls, DBCtrls,
DB, DBTables, StdCtrls, Menus;
type TE M: 1 (TE
TFmMain = class(TForm)
DBNavigator1: TDBNavigator;
GridCustomers: TDBGrid;
GridOrders: TDBGrid;
GridItems: TDBGrid;
Label1: TLabel;
Label3: TLabel;
Label4: TLabel;
MainMenu1: TMainMenu;
About1: TMenuItem;
procedure
<b>GridOrdersEnter(Sender: TObject)</b> ;
procedure
CridCustomorsEntor(Sandors

ObjectS	
الملف التنفيذي لبرنامج	
(DbErrors)	
إعداد	
علاء الدين اللباد	
واتف ۷۹۲۵۷۵۳۷۱	
program DBErrors;	
uses	

er: TObject);	TObject);
begin	procedure
if Dm.Customer.State in	GridItemsEnter(Sender: TObject);
[dsEdit,dsInsert] then	procedure
Dm.Customer.Post; {required if	GridCustomersExit(Sender:
user clicks onto details	TObject);
	procedure About1Click(Sender:
after changing key so that cascaded	TObject);
update displays	private
properly}	{ Private declarations }
end;	public
,	{ Public declarations }
procedure	end;
TFmMain.About1Click(Sender:	chu,
TObject);	var
var	FmMain: TFmMain;
fmAboutBox: TFmAboutBox;	r miviam. Trimiviam,
, and the second	implementation
begin FmAboutBox :=	implementation
	DM1 AT /
TFmAboutBox.Create(self);	uses DM1, About;
try	
FmAboutBox.showModal;	{\$R *.dfm}
finally	
FmAboutBox.free;	procedure
end;	TFmMain.GridOrdersEnter(Sender
end;	: TObject);
	begin
end.	DBNavigator1.DataSource :=
	Dm.OrdersSource;
111111111111111111111111111111111111111	end;
(*	procedure
This example represents a sampling	TFmMain.GridCustomersEnter(Sen
of the way that you might	der: TObject);
approach trapping a number of	begin
database errors.	DBNavigator1.DataSource :=
	Dm.CustomerSource;
	end;
A complete listing of the database	7
errorcodes is found in the	procedure
DBIErrs.Int file in the Delphi/Doc	TFmMain.GridItemsEnter(Sender:
directory or in the IDAPI.h	TObject);
file in the Borland Database	begin
Engine.	DBNavigator1.DataSource :=
Engine.	Dm.ItemsSource;
Database expense and defined by	end;
Database errors are defined by	ciiu,
category and code. Here's a sample:	
(EDDCAT INTECDITY)	procedure
{ ERRCAT_INTEGRITY }	TFmMain.GridCustomersExit(Send

DBIERR DETAILRECORDSEXIS	ERR
T = (ERRBASE INTEGRITY +	1; {K
ERRCODE_DETAILRECORDSEX	ERR
IST);	= 2; {
DBIERR_FORIEGNKEYERR =	ERR
(ERRBASE_INTEGRITY +	= 3; {
<pre>ERRCODE_FORIEGNKEYERR);</pre>	ERR
	4; { Fi
The ERRBASE_INTEGRITY	ERR
value is \$2600 (Hex 2600) or 9728	= 5; {
decimal.	
Thus, for example, the errorcode	ERRC
for keyviol is 9729	IST =
for master with	DELE
details is 9734.	ERR
	= 7; {
*)	incorr
	]
unit DM1;	ERRC
	=8; {
interface	range
	]
uses	ERRC
Windows, Messages, SysUtils,	RR
Classes, Graphics, Controls, Forms,	failed
Dialogs,	
DBTables, DB;	ERRC
	R =
type	failed
TDM = class(TDataModule)	EDDC
Customer: TTable;	ERRC RR
CustomerCustNo: TFloatField;	
CustomerCompany: TStringField;	Open 1 ERR
CustomerSource: TDataSource;	
Orders: TTable;	= 12; {
OrdersSource: TDataSource;	
Items: TTable;	The c
ItemsOrderNo: TFloatField;	is adde
ItemsItemNo: TFloatField;	
ItemsPartNo: TFloatField;	repres
ItemsQty: TIntegerField;	a uni
ItemsDiscount: TFloatField;	DBIF
ItemsSource: TDataSource;	(ERRI
OrdersOrderNo: TFloatField;	ERRC
OrdersCustNo: TFloatField;	DBIE
OrdersSaleDate: TDateTimeField;	(ERRI
OrdersShipDate:	ERRC
TDateTimeField;	

ERRCODE_KEYVIOL = 1; { Key violation }
ERRCODE_MINVALERR
= 2; { Min val check failed } ERRCODE MAXVALERR
= 3; { Max val check failed }
ERRCODE_REQDERR =
4; { Field value required } ERRCODE FORIEGNKEYERR
= 5; { Master record missing }
ERRCODE_DETAILRECORDSEX
IST = 6; { Cannot MODIFY or
DELETE this Master record }
ERRCODE_MASTERTBLLEVEL
= 7; { Master Table Level is incorrect }
ERRCODE LOOKUPTABLEERR
= 8; { Field value out of lookup tbl
range }
ERRCODE_LOOKUPTBLOPENE
RR = 9; { Lookup Table Open
failed }
ERRCODE DETAILTBLOPENER
$R = 10; \{0x0a \text{ Detail Table Open}\}$
failed }
ERRCODE_MASTERTBLOPENE
RR = 11; { 0x0b Master Table
Open failed }
ERRCODE_FIELDISBLANK = 12; { 0x0c Field is blank }
12, ( 0.100 1 101 10 0.11111 )
The constant for the base category
is added to these constants to
represent
a unique DBI errorcode;
DBIERR KEYVIOL =
(ERRBASE_INTEGRITY +
ERRCODE_KEYVIOL);
DBIERR_REQDERR =
(ERRBASE_INTEGRITY + ERRCODE REQUERR);
EKKCUDE_KEQDEKK);

if (E is EDBEngineError) then	OrdersEmpNo: TIntegerField;
if (E as	procedure
EDBEngineError).Errors[0].Errorc	CustomerPostError(DataSet:
ode = eKeyViol then	TDataSet; E: EDatabaseError;
begin	var Action: TDataAction);
MessageDlg('Unable to post:	procedure
<b>Duplicate Customer ID.'</b> ,	CustomerDeleteError(DataSet:
mtWarning, [mbOK], 0);	TDataSet; E: EDatabaseError;
Abort;	var Action: TDataAction);
end;	procedure
end;	ItemsPostError(DataSet: TDataSet;
,	E: EDatabaseError;
procedure	var Action: TDataAction);
TDM.CustomerDeleteError(DataSet	procedure
: TDataSet;	OrdersPostError(DataSet:
E: EDatabaseError; var Action:	TDataSet; E: EDatabaseError;
TDataAction);	var Action: TDataAction);
begin	procedure
if (E is EDBEngineError) then	OrdersDeleteError(DataSet:
if (E as	TDataSet; E: EDatabaseError;
EDBEngineError).Errors[0].Errorc	var Action: TDataAction);
ode = eDetailsExist then	private
{the customer record has	{ Private declarations }
dependent details in the Orders	public
table.}	{ Public declarations }
begin	end;
MessageDlg('To delete this	Cital
record, first delete related orders	var
and items.',	DM: TDM;
mtWarning, [mbOK], 0);	
Abort;	const
end;	{Declare constants we're interested
end;	in}
	eKeyViol = 9729;
procedure	eRequiredFieldMissing = 9732;
TDM.ItemsPostError(DataSet:	eForeignKey = 9733;
TDataSet; E: EDatabaseError;	eDetailsExist = 9734;
var Action: TDataAction);	CDCtailsEalst = 7/37,
begin	
This error will occur when a part	implementation
number is specified that	пириспецион
is not in the parts table.}	{\$R *.dfm}
if (E as	this said
EDBEngineError).Errors[0].Errorc	procedure
ode = eForeignKey then	procedure TDM CustomorPostFrror(DataSate
begin	TDM.CustomerPostError(DataSet: TDataSet;
MessageDlg('Part number is	,
invalid', mtWarning,[mbOK],0);	E: EDatabaseError; var Action: TDataAction);
Abort;	
1100109	begin

and re	lated items?',
	nfirmation,
[n	mbYes, mbNo], 0) = mrYes
then	
beg	gin
{I	Delete related records in linked
'items'	' table}
w	hile Items.RecordCount > 0 do
	Items.delete;
{ <b>I</b>	Finally,delete this record}
	ction := daRetry;
enc	d else Abort;
end	<u>,                                      </u>
end;	
end.	
\\\\\\	
unit A	bout:
_	
linterfa	ice
interfa	ace
uses W	Vindows, SysUtils, Classes,
uses W Graph	Vindows, SysUtils, Classes, nics, Forms, Controls, StdCtrls,
uses W Graph	Vindows, SysUtils, Classes,
uses W Graph Butto	Vindows, SysUtils, Classes, nics, Forms, Controls, StdCtrls,
uses W Graph Butto	Vindows, SysUtils, Classes, nics, Forms, Controls, StdCtrls, ons, ExtCtrls, ComCtrls;
uses W Graph Butto type TFm.	Vindows, SysUtils, Classes, cics, Forms, Controls, StdCtrls, ons, ExtCtrls, ComCtrls;  AboutBox = class(TForm)
uses W Graph Butto type TFm.	Vindows, SysUtils, Classes, cics, Forms, Controls, StdCtrls, cons, ExtCtrls, ComCtrls;  AboutBox = class(TForm) el1: TPanel;
uses W Graph Butto type TFm Pan Prog	Vindows, SysUtils, Classes, nics, Forms, Controls, StdCtrls, ons, ExtCtrls, ComCtrls;  AboutBox = class(TForm) el1: TPanel; gramIcon: TImage;
uses W Graph Butto type TFm. Pan Prog	Vindows, SysUtils, Classes, cics, Forms, Controls, StdCtrls, cons, ExtCtrls, ComCtrls;  AboutBox = class(TForm) el1: TPanel; gramIcon: TImage; ductName: TLabel;
type TFm. Pan Proc	Vindows, SysUtils, Classes, nics, Forms, Controls, StdCtrls, ons, ExtCtrls, ComCtrls;  AboutBox = class(TForm) el1: TPanel; gramIcon: TImage; ductName: TLabel; sion: TLabel;
uses W Graph Butto type TFm. Pan Prog Prog Vers	Vindows, SysUtils, Classes, nics, Forms, Controls, StdCtrls, ons, ExtCtrls, ComCtrls;  AboutBox = class(TForm) el1: TPanel; gramIcon: TImage; ductName: TLabel; sion: TLabel; el2: TPanel;
type TFm. Pan Prog	Vindows, SysUtils, Classes, nics, Forms, Controls, StdCtrls, ons, ExtCtrls, ComCtrls;  AboutBox = class(TForm) el1: TPanel; gramIcon: TImage; ductName: TLabel; sion: TLabel; el2: TPanel; el2: TLabel;
type TFm. Pan Prog Prod Vers Pan Lab	Vindows, SysUtils, Classes, nics, Forms, Controls, StdCtrls, ons, ExtCtrls, ComCtrls;  AboutBox = class(TForm) el1: TPanel; gramIcon: TImage; ductName: TLabel; sion: TLabel; el2: TPanel; el2: TLabel; ge1: TImage;
type TFm. Pan Prog Prog Lab Ima	Vindows, SysUtils, Classes, cics, Forms, Controls, StdCtrls, cons, ExtCtrls, ComCtrls;  AboutBox = class(TForm) el1: TPanel; gramIcon: TImage; ductName: TLabel; sion: TLabel; el2: TPanel; el2: TLabel; ge1: TImage; no1: TMemo;
type TFm. Pan Prog Prog Vers Lab Ima Mer Butte	Vindows, SysUtils, Classes, nics, Forms, Controls, StdCtrls, ons, ExtCtrls, ComCtrls;  AboutBox = class(TForm) el1: TPanel; gramIcon: TImage; ductName: TLabel; sion: TLabel; el2: TPanel; el2: TLabel; ge1: TImage; no1: TMemo; ton1: TButton;
uses W Graph Butto type TFm. Pan Prog Prog Vers Pan Lab Ima Mer Butto	Vindows, SysUtils, Classes, nics, Forms, Controls, StdCtrls, ons, ExtCtrls, ComCtrls;  AboutBox = class(TForm) el1: TPanel; gramIcon: TImage; ductName: TLabel; sion: TLabel; el2: TPanel; el2: TPanel; el2: TLabel; ge1: TImage; no1: TMemo; ton1: TButton; tte
uses W Graph Butto  type TFm. Pan Prog Prog Vers Pan Lab Ima Mer Butto	Vindows, SysUtils, Classes, nics, Forms, Controls, StdCtrls, ons, ExtCtrls, ComCtrls;  AboutBox = class(TForm) el1: TPanel; gramIcon: TImage; ductName: TLabel; sion: TLabel; el2: TPanel; el2: TLabel; ge1: TImage; no1: TMemo; ton1: TButton;
uses W Graph Butto  type TFm. Pan Prog Prog Vers Pan Lab Ima Mer Butto	Vindows, SysUtils, Classes, nics, Forms, Controls, StdCtrls, ons, ExtCtrls, ComCtrls;  AboutBox = class(TForm) el1: TPanel; gramIcon: TImage; ductName: TLabel; sion: TLabel; el2: TPanel; el2: TPanel; el2: TLabel; ge1: TImage; no1: TMemo; ton1: TButton; te ivate declarations } c
uses W Graph Butto  type TFm. Pan Prog Prog Vers Pan Lab Ima Mer Butto	Vindows, SysUtils, Classes, nics, Forms, Controls, StdCtrls, ons, ExtCtrls, ComCtrls;  AboutBox = class(TForm) el1: TPanel; gramIcon: TImage; ductName: TLabel; sion: TLabel; el2: TPanel; el2: TLabel; ge1: TImage; no1: TMemo; ton1: TButton; te ivate declarations }
uses W Graph Butto  type TFm. Pan Prog Prog Vers Pan Lab Ima Mer Butto	Vindows, SysUtils, Classes, nics, Forms, Controls, StdCtrls, ons, ExtCtrls, ComCtrls;  AboutBox = class(TForm) el1: TPanel; gramIcon: TImage; ductName: TLabel; sion: TLabel; el2: TPanel; el2: TPanel; el2: TLabel; ge1: TImage; no1: TMemo; ton1: TButton; te ivate declarations } c
uses W Graph Butto  type TFm. Pan Prog Prog Vers Pan Lab Ima Mer Butt priva {Pr	Vindows, SysUtils, Classes, nics, Forms, Controls, StdCtrls, ons, ExtCtrls, ComCtrls;  AboutBox = class(TForm) el1: TPanel; gramIcon: TImage; ductName: TLabel; sion: TLabel; el2: TPanel; el2: TPanel; el2: TLabel; ge1: TImage; no1: TMemo; ton1: TButton; te ivate declarations } c
uses W Graph Butto  type TFm. Pan Prog Prog Vers Pan Lab Ima Mer Butt priva { Pr	Vindows, SysUtils, Classes, nics, Forms, Controls, StdCtrls, ons, ExtCtrls, ComCtrls;  AboutBox = class(TForm) el1: TPanel; gramIcon: TImage; ductName: TLabel; sion: TLabel; el2: TPanel; el2: TPanel; el2: TLabel; ge1: TImage; no1: TMemo; ton1: TButton; te ivate declarations } c
uses W Graph Butto  type TFm. Pan Prog Prog Vers Pan Lab Ima Mer Butt priva {Pr publi {Pu end;	Vindows, SysUtils, Classes, nics, Forms, Controls, StdCtrls, ons, ExtCtrls, ComCtrls;  AboutBox = class(TForm) el1: TPanel; gramIcon: TImage; ductName: TLabel; sion: TLabel; el2: TPanel; el2: TPanel; el2: TLabel; ge1: TImage; no1: TMemo; ton1: TButton; te ivate declarations } c
uses W Graph Butto  type TFm. Pan Prog Prog Vers Pan Lab Ima Mer Butt priva {Pr publi {Pu end;	Vindows, SysUtils, Classes, nics, Forms, Controls, StdCtrls, ons, ExtCtrls, ComCtrls;  AboutBox = class(TForm) el1: TPanel; gramIcon: TImage; ductName: TLabel; sion: TLabel; el2: TPanel; el2: TLabel; ge1: TImage; no1: TMemo; ton1: TButton; te ivate declarations } c ablic declarations }
uses W Graph Butto  type TFm. Pan Prog Prog Vers Pan Lab Ima Mer Butt priva {Pr publi {Pu end;	Vindows, SysUtils, Classes, nics, Forms, Controls, StdCtrls, ons, ExtCtrls, ComCtrls;  AboutBox = class(TForm) el1: TPanel; gramIcon: TImage; ductName: TLabel; sion: TLabel; el2: TPanel; el2: TLabel; ge1: TImage; no1: TMemo; ton1: TButton; te ivate declarations } c ablic declarations }

end;
end;
procedure
TDM.OrdersPostError(DataSet:
TDataSet; E: EDatabaseError;
var Action: TDataAction);
var
iDBIError: Integer;
begin
if (E is EDBEngineError) then
begin
iDBIError := (E as
EDBEngineError).Errors[0].Errorc
ode;
case iDBIError of
eRequiredFieldMissing:
The EmpNo field is defined as
being required.
begin Massaga Dig ('Plaasa massida an
MessageDlg('Please provide an
Employee ID', mtWarning, [mbOK],
0);
Abort;
end;
eKeyViol:
{The primary key is OrderNo}
begin
MessageDlg('Unable to post.
<b>Duplicate Order Number'</b> ,
mtWarning,
[mbOK], 0);
Abort;
end;
end;
end;
end;
procedure
TDM.OrdersDeleteError(DataSet:
TDataSet; E: EDatabaseError;
var Action: TDataAction);
begin
if E is EDBEngineError then
if (E as
EDBEngineError).Errors[0].Errorc
ode = eDetailsExist then
begin
if MessageDlg('Delete this order
ii iiicssageDig( Delete tills of del

Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
object Label3: TLabel
Left = 8
Top = 121 Width = 48
Height = 16
Caption = 'Orders'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
object Label4: TLabel
Left = 8
Top = 209
Width = 38
Height = 16
Caption = 'Items'
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
object DBNavigator1:
TDBNavigator
Left = 5
Top = 4 Width = 240
Height = 25
TabOrder = 0
end
object GridCustomers: TDBGrid
Left = 8
Top = 47
Width = 342
Height = 73
DataSource =
DM.CustomerSource

{\$R *.dfm}
end.

ObjectS	
الهلف النصي لبرناهج	
(DbErrors14)	
إعداد	
علاء الدين اللباد	
واته ۷۳۵۷۵۶۶۰	

object FmMain: TFmMain
Left = 280
Top = 286
BorderStyle = bsSingle
Caption = 'Database Errors Demo'
ClientHeight = 352
ClientWidth = 359
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
Menu = MainMenu1
OldCreateOrder = True
PixelsPerInch = 96
TextHeight = 13
object Label1: TLabel
Left = 8
Top = 32
Width = 74
Height = 16
Caption = 'Customers'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText

EigldNama - !CalaData!
FieldName = 'SaleDate'
Visible = True
end
item
Expanded = False
FieldName = 'ShipDate'
Visible = True
end
item
Expanded = False
FieldName = 'EmpNo'
Width = 47
Visible = True
end>
end
object GridItems: TDBGrid
Left = 8
Top = 224
Width = 341
Height = 120
DataSource = DM.ItemsSource
TabOrder = 3
TitleFont.Charset =
DEFAULT CHARSET
TitleFont.Color = clWindowText
TitleFont.Height = -11
TitleFont.Name = 'MS Sans Serif'
TitleFont.Style = []
OnEnter = GridItemsEnter
end
object MainMenu1: TMainMenu
Left = 296
object About1: TMenuItem
Caption = '&About'
OnClick = About1Click
end
end
End
Ditt
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
object Em About Day: TEm About Day
object FmAboutBox: TFmAboutBox Left = 224
Top = 279
BorderStyle = bsDialog
Caption = 'About'
ClientHeight = 403
ClientWidth = 419

TabOrder = 1
TitleFont.Charset =
DEFAULT_CHARSET
TitleFont.Color = clWindowText
TitleFont.Height = -11
TitleFont.Name = 'MS Sans Serif'
TitleFont.Style = []
OnEnter = GridCustomersEnter
OnExit = GridCustomersExit
Columns = <
item
Expanded = False
FieldName = 'CustNo'
Visible = True
end
item
Expanded = False
FieldName = 'Company'
Visible = True
end>
V == V-
end TEDEC : 1
object GridOrders: TDBGrid
Left = 8
Top = 136
Width = 341
Height = 75
DataSource = DM.OrdersSource
TabOrder = 2
TitleFont.Charset =
DEFAULT_CHARSET
TitleFont.Color = clWindowText
TitleFont.Height = -11
TitleFont.Name = 'MS Sans Serif'
TitleFont.Style = []
OnEnter = GridOrdersEnter
Columns = <
item
Expanded = False
FieldName = 'OrderNo'
Visible = True
end
item
Expanded = False
FieldName = 'CustNo'
Visible = True
end
item
Expanded = False

1.4 . B 14 EB 1
object Panel2: TPanel
Left = 9
Top = 88
Width = 401
Height = 73
TabOrder = 1
object Label2: TLabel
Left = 22
Top = 8
Width = 194
Height = 13
Caption = 'Data Model -
Referential Integrity'
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
object Image1: TImage
Left = 18
Top = 24
Width = 325
Height = 46
Picture.Data = {
}
end
end
object Memo1: TMemo
Left = 9
Top = 168
Width = 401
Height = 193
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
Lines.Strings = (
'This example demonstrates'
' - The use of Data Modules to
centralize coding'
' - A one-to-many-to-many form'
' - Trapping and Controlling

Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
PixelsPerInch = 96
TextHeight = 13
object Panel1: TPanel
Left = 8
Top = 8
Width = 402
Height = 73
BevelInner = bvRaised
BevelOuter = bvLowered
TabOrder = 0
object ProgramIcon: TImage
Left = 8
Top = 8
Width = 65
Height = 57
Picture.Data = {
0000
0000}
Stretch = True
IsControl = True
end TV I I
object ProductName: TLabel
Left = 88
Top = 16
Width = 119
Height = 13
Caption = 'Database Errors
Example'
IsControl = True
end
object Version: TLabel
Left = 88
Top = 40
Width = 53
Height = 13
Caption = 'Version 1.0'
IsControl = True
end
end
ena

'General Notes:'	<b>Database Errors'</b>
"	"
' The code in this example for the	'* Creation Order'
OnDeleteError on the Orders '	"
	'The proper functioning of any
' table allows you to delete the	form with a one-to-many '
current Orders record even if i' +	'relationship depends on the
't '	creation order of TTables and '
' has dependent details. If this	'TDatasources at runtime. In this
error occurs, a dialog box'	example, the MasterSource '
' asks you if you want to remove	'property of the Orders table
the Orders record and all'	point to CustomerSource. For the '
	'one-to-many link to properly
' of its details. If this is action	synch up, the Customer table must '
confirmed, the code deletes' +	'be active. The creation order is
'all'	determined at design'
' of the related Items records	'time by right-clicking on the
and then completes the action'	form and choosing Creation Order.'
' by retrying the deletion for the	11
current record.	'* Databasa Errar Tranning'
current record.	'* Database Error Trapping'
1. The On Deed France	Delahi disalara amana that arisa
' The OnPostError,	'Delphi displays errors that arise
OnEditError and OnDeleteError	from user interaction with '
pass to you'	'database tables as application
' an exception (e) of the	level exceptions. The new '
EDatabaseError class. To get '	'OnPostError, OnEditError and
' engine-level errorcodes, you	OnDeleteError events of '
must cast this as an error of	'the TTable and TQuery objects
' type EDBEngineError. See the	allow you to trap these '
code in DM.PAS for details.'	'errors closer to their origin. '
"	**
' The CustNo *Tfield* has min	
and max values. These are'	'To trigger these errors, run
' NOT database engine-level Min	this application, and try to modify' +
and Max Valchecks and so'	','
' errors that arise from invalid	'add, or delete records. '
entries are not caught by the '	"
' engine errors trapped for	'You may encounter several
here.')	different database engine'
ParentFont = False	'errors: key violations, missing
ReadOnly = True	required fields, referential'
ScrollBars = ssVertical	'integrity errors, incorrect
TabOrder = 2	foreign keys (most notably from '
WantReturns = False	
end	'the "parts.db" table, which is
object Button1: TButton	not explicitly in this applicatio' +
Left = 173	'n), '
Top = 368	'and so on. '
Width = 75	11
	<u> </u>

DataSet = Customer
Left = 81
Top = 8
end
object Orders: TTable
Active = True
OnDeleteError =
OrdersDeleteError
OnPostError = OrdersPostError
DatabaseName = 'DBDEMOS'
IndexName = 'CustNo'
MasterFields = 'CustNo'
MasterSource = CustomerSource
TableName = 'ORDERS.DB'
Left = 17
Top = 70
object OrdersOrderNo:
TFloatField
DisplayWidth = 10
FieldName = 'OrderNo'
DisplayFormat = '''#''0000'
EditFormat = '0000'
end
object OrdersCustNo: TFloatField
Alignment = taLeftJustify
FieldName = 'CustNo'
DisplayFormat = 'CN 0000'
EditFormat = '0000'
MaxValue =
9999.000000000000000000
MinValue =
1000.000000000000000000
end
object OrdersSaleDate:
TDateTimeField
FieldName = 'SaleDate'
end
object OrdersShipDate:
<b>TDateTimeField</b>
FieldName = 'ShipDate'
FieldName = 'ShipDate' end
FieldName = 'ShipDate' end object OrdersEmpNo:
FieldName = 'ShipDate' end object OrdersEmpNo: TIntegerField
FieldName = 'ShipDate' end object OrdersEmpNo: TIntegerField FieldName = 'EmpNo'
FieldName = 'ShipDate' end object OrdersEmpNo: TIntegerField FieldName = 'EmpNo' DisplayFormat = 'Emp''#'' 0000'
FieldName = 'ShipDate' end object OrdersEmpNo: TIntegerField FieldName = 'EmpNo' DisplayFormat = 'Emp''#'' 0000' EditFormat = '0000'
FieldName = 'ShipDate' end object OrdersEmpNo: TIntegerField FieldName = 'EmpNo' DisplayFormat = 'Emp''#'' 0000'

Height = 25
Cancel = True
Caption = 'OK'
Default = True
ModalResult = 2
TabOrder = 3
end
End
Enu
111111111111111111111111111111111111111
11 . D. ( ) D. (
object DM: TDM
OldCreateOrder = True
Left = 347
Top = 307
Height = 242
Width = 222
object Customer: TTable
Active = True
OnDeleteError =
CustomerDeleteError
OnPostError =
CustomerPostError
DatabaseName = 'dbdemos'
TableName = 'CUSTOMER.DB'
Left = 17
Top = 8
object CustomerCustNo:
TFloatField
Alignment = taLeftJustify
DisplayWidth = 7
FieldName = 'CustNo'
DisplayFormat = 'CN 0000'
EditFormat = '0000'
MaxValue =
9999.000000000000000000
MinValue =
1000.0000000000000000000
end
0-1-0-
object CustomerCompany:
object CustomerCompany: TStringField
object CustomerCompany: TStringField DisplayWidth = 22
object CustomerCompany: TStringField DisplayWidth = 22 FieldName = 'Company'
object CustomerCompany: TStringField DisplayWidth = 22 FieldName = 'Company' Size = 30
object CustomerCompany: TStringField DisplayWidth = 22 FieldName = 'Company' Size = 30 end
object CustomerCompany: TStringField DisplayWidth = 22 FieldName = 'Company' Size = 30 end end
object CustomerCompany: TStringField DisplayWidth = 22 FieldName = 'Company' Size = 30 end

Top = 13	30
end	
End	
\\\\\	

Procedure860
الهلف التنفيذي لبرناهج
(Activefm)
إعداد
علاء الدين اللباد
واتف ۷۳۷۱۹۹۹
library EmpEditX;
{ This project demonstrates using client datasets in an ActiveForm.
The ActiveForm in this project works just like the client in the
EMPEDIT demo. Before compiling
and using this Active Library you should compile and run the Server
project from the EMPEDIT demo.
For infomation on deploying
ActiveForms, look for "ActiveX controls:deploying" in
the online help index.
}
uses
ComServ,
EmpE_TLB in 'EmpE_TLB.pas',
EmpEdImp in 'EmpEdImp.pas'
{EmpEditForm: TActiveForm}
{EmpEditForm: CoClass},
recerror in
'\\objrepos\recerror.pas';
{\$E ocx}
exports
DllGetClassObject,
DllCanUnloadNow,
DllRegisterServer,

end
end
object OrdersSource: TDataSource
DataSet = Orders
Left = 81
Top = 70
end
object Items: TTable
Active = True
OnPostError = ItemsPostError
DatabaseName = 'DBDEMOS'
IndexFieldNames = 'OrderNo'
MasterFields = 'OrderNo'
MasterFields OrderNo  MasterSource = OrdersSource
TableName = 'ITEMS.DB'
Left = 17
Top = 130
object ItemsOrderNo: TFloatField
DisplayWidth = 10
FieldName = 'OrderNo'
DisplayFormat = '0000'
end
object ItemsItemNo: TFloatField
DisplayWidth = 10
FieldName = 'ItemNo'
end
object ItemsPartNo: TFloatField
Alignment = taLeftJustify
DisplayWidth = 10
FieldName = 'PartNo'
DisplayFormat = 'PN-00000'
EditFormat = '00000'
end
object ItemsQty: TIntegerField
DisplayWidth = 6
FieldName = 'Qty'
end
object ItemsDiscount: TFloatField
DisplayWidth = 10
FieldName = 'Discount'
DisplayFormat = '#%'
MinValue =
100.00000000000000000
end
end
object ItemsSource: TDataSource
DataSet = Items
Left = 81

II I A D AA CH I AC I	Duri
UpdateButtonClick(Sender:	DllUnregisterServer;
TObject);	
<> procedure	{\$R *.TLB}
UndoButtonClick(Sender: TObject);	
<> procedure	{\$R *.RES}
EmployeesReconcileError(DataSet:	
TCustomClientDataSet;	begin
E: EReconcileError;	end.
UpdateKind: TUpdateKind; var	
Action: TReconcileAction);	
<> procedure	\\\\\\\
EmpDataDataChange(Sender:	
TObject; Field: TField);	unit amnadimn
private	unit empedimp;
{ Private declarations }	• , , , ,
FEvents: IEmpEditFormEvents;	interface
<> procedure	
ActivateEvent(Sender: TObject);	uses
	Windows, Messages, SysUtils,
<pre>&lt;&gt; procedure ClickEvent(Sender, TObiest);</pre>	Classes, Graphics, Controls, Forms,
ClickEvent(Sender: TObject);	Dialogs,
<> procedure	ActiveX, AxCtrls, EmpE_TLB,
CreateEvent(Sender: TObject);	DBClient, Db, ComCtrls, ExtCtrls,
<> procedure	DBCtrls, StdCtrls, Mask,
DblClickEvent(Sender: TObject);	MConnect;
<> procedure	
DeactivateEvent(Sender: TObject);	type
<> procedure	TEmpEditForm =
DestroyEvent(Sender: TObject);	class(TActiveForm, IEmpEditForm)
<> procedure	Label2: TLabel;
KeyPressEvent(Sender: TObject;	UpdateButton: TButton;
var Key: Char);	UndoButton: TButton;
<> procedure	QueryButton: TButton;
PaintEvent(Sender: TObject);	DBText1: TDBText;
protected	FirstName: TDBEdit;
{ Protected declarations }	,
<> procedure	LastName: TDBEdit;
EventSinkChanged(const	PhoneExt: TDBEdit;
EventSink: IUnknown); override;	HireDate: TDBEdit;
<> procedure Initialize;	Salary: TDBEdit;
override;	EmpData: TDataSource;
function Get Active: WordBool;	DBNavigator1: TDBNavigator;
safecall;	Employees: TClientDataSet;
function Get AutoScroll:	MidasConnection:
WordBool; safecall;	TDCOMConnection;
function Get AxBorderStyle:	RecInd: TLabel;
TxActiveFormBorderStyle; safecall;	<> procedure
	QueryButtonClick(Sender:
function Get_Caption:	TOI: A
WidoStwings safaacille	TObject);
WideString; safecall; function Get Color: OLE Color;	<pre>TObject); &lt;&gt; procedure</pre>

Z	<b>c</b>
<> procedure	safecall;
Set_KeyPreview(Value: WordBool);	function Get_Cursor: Smallint;
safecall;	safecall;
<> procedure	function Get_DropTarget:
Set_PixelsPerInch(Value: Integer);	WordBool; safecall;
safecall;	function Get_Enabled: WordBool;
<> procedure	safecall;
Set_PrintScale(Value:	function Get_Font: Font; safecall;
TxPrintScale); safecall;	function Get HelpFile:
<> procedure	WideString; safecall;
Set_Scaled(Value: WordBool);	function Get KeyPreview:
safecall;	WordBool; safecall;
<> procedure	function Get PixelsPerInch:
Set Visible(Value: WordBool);	Integer; safecall;
safecall;	function Get PrintScale:
<> procedure	TxPrintScale; safecall;
Set WindowState(Value:	function Get Scaled: WordBool;
TxWindowState); safecall;	safecall;
public	function Get Visible: WordBool;
{ Public declarations }	safecall;
end;	function Get WindowState:
	TxWindowState; safecall;
implementation	<> procedure
Implementation	Set AutoScroll(Value: WordBool);
uses RecError, ComServ;	safecall;
uses Received, Comserv,	,
(CD * df)	<> procedure
{\$R *.dfm}	Set_AxBorderStyle(Value:
(TPE 12 1/4E )	TxActiveFormBorderStyle); safecall;
{ TEmpEditForm }	,
	<> procedure
<> procedure	Set_Caption(const Value:
TEmpEditForm.EventSinkChanged(	WideString); safecall;
const EventSink: IUnknown);	<> procedure
begin	Set_Color(Value: OLE_Color);
FEvents := EventSink as	safecall;
IEmpEditFormEvents;	<> procedure
end;	Set_Cursor(Value: Smallint);
	safecall;
<> procedure	<> procedure
TEmpEditForm.Initialize;	Set_DropTarget(Value: WordBool);
begin	safecall;
OnActivate := ActivateEvent;	<> procedure
OnClick := ClickEvent;	Set_Enabled(Value: WordBool);
OnCreate := CreateEvent;	safecall;
OnDblClick := DblClickEvent;	<> procedure Set_Font(const
OnDeactivate := DeactivateEvent;	Value: Font); safecall;
OnDestroy := DestroyEvent;	<> procedure
OnKeyPress := KeyPressEvent;	Set_HelpFile(const Value:
OnPaint := PaintEvent;	WideString); safecall;
On and and and vent,	

end;	end;
	,
function	function
TEmpEditForm.Get_Enabled:	TEmpEditForm.Get_Active:
WordBool;	WordBool;
begin	begin
Result := Enabled;	Result := Active;
end;	end;
function TEmpEditForm.Get_Font:	function
Font;	TEmpEditForm.Get_AutoScroll:
begin	WordBool;
GetOleFont(Font, Result);	begin
end;	Result := AutoScroll;
	end;
function	6 4
TEmpEditForm.Get_HelpFile:	function
WideString;	TEmpEditForm.Get_AxBorderStyle
begin	: TxActiveFormBorderStyle;
Result := WideString(HelpFile);	begin R 1 (A R 1 S( I )
end;	Result := Ord(AxBorderStyle);
6	end;
function	6 4
TEmpEditForm.Get_KeyPreview:	function
WordBool;	TEmpEditForm.Get_Caption:
begin	WideString; begin
Result := KeyPreview;	
end;	Result := WideString(Caption); end;
function	enu,
TEmpEditForm.Get PixelsPerInch:	function
Integer;	TEmpEditForm.Get Color:
begin	OLE COLOR;
Result := PixelsPerInch;	begin
end;	Result := Color;
	end;
function	/
TEmpEditForm.Get PrintScale:	function
TxPrintScale;	TEmpEditForm.Get Cursor:
begin	Smallint;
Result := Ord(PrintScale);	begin
end;	Result := Smallint(Cursor);
,	end;
function	
TEmpEditForm.Get Scaled:	function
WordBool;	TEmpEditForm.Get_DropTarget:
begin	WordBool;
Result := Scaled;	begin
end;	Result := DropTarget;

end;	
	function
<> procedure	TEmpEditForm.Get_Visible:
TEmpEditForm.Set_DropTarget(Va	WordBool;
lue: WordBool);	begin
begin	Result := Visible;
	·
DropTarget := Value;	end;
end;	£
	function
<> procedure	TEmpEditForm.Get_WindowState:
TEmpEditForm.Set_Enabled(Value:	TxWindowState;
WordBool);	begin
begin	Result := Ord(WindowState);
Enabled := Value;	end;
end;	
	<> procedure
<> procedure	TEmpEditForm.Set_AutoScroll(Val
TEmpEditForm.Set_Font(const	ue: WordBool);
Value: Font);	begin
begin	AutoScroll := Value;
SetOleFont(Font, Value);	end;
end;	
	<> procedure
<> procedure	TEmpEditForm.Set_AxBorderStyle(
TEmpEditForm.Set_HelpFile(const	Value: TxActiveFormBorderStyle);
Value: WideString);	begin
begin	AxBorderStyle :=
HelpFile := String(Value);	TActiveFormBorderStyle(Value);
end;	end;
<> procedure	<> procedure
TEmpEditForm.Set_KeyPreview(Va	TEmpEditForm.Set_Caption(const
lue: WordBool);	Value: WideString);
begin	begin
KeyPreview := Value;	Caption := TCaption(Value);
end;	end;
CHU,	· · · · · · · · · · · · · · · · · · ·
<> procedure	<> procedure
TEmpEditForm.Set_PixelsPerInch(	TEmpEditForm.Set_Color(Value:
Value: Integer);	OLE COLOR);
begin	begin
PixelsPerInch := Value;	Color := Value;
· · · · · · · · · · · · · · · · · · ·	end;
end;	cnu,
Z Nama and June	z > nwoooduwc
<> procedure	<pre>&lt;&gt; procedure TEmpEditEonm Set Congor(Volume)</pre>
TEmpEditForm.Set_PrintScale(Val	TEmpEditForm.Set_Cursor(Value:
ue: TxPrintScale);	Smallint);
begin	begin
<b>PrintScale</b> := TPrintScale(Value);	Cursor := TCursor(Value);

	_
TEmpEditForm.DblClickEvent(Sen	end;
der: TObject);	
begin	<> procedure
if FEvents <> nil then	TEmpEditForm.Set_Scaled(Value:
FEvents.OnDblClick;	WordBool);
end;	begin
	Scaled := Value;
<> procedure	end;
TEmpEditForm.DeactivateEvent(Se	,
nder: TObject);	<> procedure
begin	TEmpEditForm.Set_Visible(Value:
if FEvents <> nil then	WordBool);
FEvents.OnDeactivate;	begin
end;	Visible := Value;
enu,	,
Z Nama and una	end;
<> procedure	
TEmpEditForm.DestroyEvent(Send	<> procedure
er: TObject);	TEmpEditForm.Set_WindowState(
begin	Value: TxWindowState);
if FEvents <> nil then	begin
FEvents.OnDestroy;	WindowState :=
end;	TWindowState(Value);
	end;
<> procedure	
TEmpEditForm.KeyPressEvent(Sen	<> procedure
der: TObject; var Key: Char);	TEmpEditForm.ActivateEvent(Send
var	er: TObject);
TempKey: Smallint;	begin
begin	if FEvents <> nil then
TempKey := Smallint(Key);	FEvents.OnActivate;
if FEvents <> nil then	end;
FEvents.OnKeyPress(TempKey);	chu,
Key := Char(TempKey);	<> procedure
end;	TEmpEditForm.ClickEvent(Sender:
cnu,	TObject);
< > nrogoduro	begin
<> procedure TEmpEditEarm PointEvent(Sandon)	if FEvents <> nil then
TEmpEditForm.PaintEvent(Sender:	
TObject);	FEvents.OnClick;
begin	end;
if FEvents <> nil then	
FEvents.OnPaint;	<> procedure
end;	TEmpEditForm.CreateEvent(Sende
	r: TObject);
{	begin
	if FEvents <> nil then
	FEvents.OnCreate;
}	end;
<> procedure	<> procedure

association with the change. }	TEmpEditForm.QueryButtonClick(
	Sender: TObject);
	begin
Employees.UndoLastChange(True);	{ Get data from the server. The
	number of records returned is
end;	dependent on
	the PacketRecords property of
<> procedure	TClientDataSet. If PacketRecords is
TEmpEditForm.EmployeesReconcil	set
eError(DataSet:	to -1 then all records are returned.
TCustomClientDataSet;	Otherwise, as the user scrolls
E: EReconcileError; UpdateKind:	through the data, additional
• •	records are returned in separate
TUpdateKind; var Action:	data packets. }
TReconcileAction);	data packets. }
begin	E I CI
	Employees.Close;
{ This is the event handler which is	Employees.Open;
called when there are errors during	
the	end;
update process. To demonstrate,	
you can create an error by running	<> procedure
two	TEmpEditForm.UpdateButtonClick
copies of this application and	(Sender: TObject);
modifying the same record in each	begin
one.	9
Here we use the standard reconcile	{ Apply any edits. The parameter
error dialog from the object	indicates the number of errors which
repository. }	are allowed before the updating is
	aborted. A value of -1 indicates that
Action :=	all successful updates be applied
HandleReconcileError(DataSet,	regardless of the number of errors.
UpdateKind, E);	regardless of the number of errors.
end;	F
chu,	Employees.ApplyUpdates(-1);
<> procedure	end;
TEmpEditForm.EmpDataDataChan	enu,
ge(Sender: TObject; Field: TField);	
	<> procedure
begin	TEmpEditForm.UndoButtonClick(S
	ender: TObject);
{ This code is used to update the	begin
status bar to show the number of	
records	{ Here we demonstrate a new feature
that have been retrieved an our	in TClientDataSet, the ability to
relative position with the dataset. }	undo
	changes in reverse order. The
with Employees do	parameter indicates if the cursor
if Active then	should
RecInd.Caption := Format(' %d	be repositioned to the record

<> Object EmpEditForm:
TEmpEditForm
Left = 219
Top = 138
Width = 453
Height = 212
Caption = 'Employee
Administrator'
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'Default'
Font.Style = []
OldCreateOrder = True
PixelsPerInch = 96
TextHeight = 13
<> Object Label2: TLabel
Left = 28
Top = 61
Width = 50
Height = 13
Caption = 'First Name'
FocusControl = FirstName
end
<> Object Label3: TLabel
Left = 123
Top = 61
Width = 61
Height = 13
AutoSize = False
Caption = 'Last Name'
FocusControl = LastName
end
<> Object Label4: TLabel
Left = 240
Top = 62
Width = 15
Height = 13
Caption = 'Ext'
FocusControl = PhoneExt
end
<> Object Label5: TLabel
Left = 28
Top = 118
10h - 110

of %d', [RecNo, R	ecordCount]);
end;	
initialization	
TActiveFormFac	ctory.Create(
ComServer,	
TActiveFormCo	ontrol,
TEmpEditForn	ı,
Class_EmpEdit	Form,
1,	
",	
OLEMISC_SIN	1PLEFRAME or
OLEMISC_ACTS	SLIKELABEL);
end.	
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	.\\\\\\

<> Object\$22
الهلف النصي لبرناهج
(Activefm)
إعداد
علاء الدين اللباد
واتف ۷۳۵۷۵۶۹۰

TDBEdit		
Left = 28		
Top = 76		
Width = 78		
Height = 21		
DataField = 'FirstName'		
DataSource = EmpData		
TabOrder = 0		
end		
<> Object LastName:		
TDBEdit		
Left = 123		
Top = 76		
Width = 99		
Height = 21		
DataField = 'LastName'		
DataSource = EmpData		
TabOrder = 1		
end		
<> Object PhoneExt: TDBEdit		
Left = 240		
Top = 76		
Width = 49		
Height = 21		
DataField = 'PhoneExt'		
DataSource = EmpData		
TabOrder = 2		
end		
<> Object HireDate: TDBEdit		
Left = 28		
Top = 132		
Width = 73		
Height = 21		
DataField = 'HireDate'		
DataSource = EmpData		
TabOrder = 3		
end		
<> Object Salary: TDBEdit		
Left = 121		
Top = 132		
Width = 96		
Height = 21		
DataField = 'Salary'		
DataSource = EmpData		
TabOrder = 4		
end		
<> Object QueryButton:		
TButton		

Width = 45		
Height = 13		
Caption = 'Hire Date'		
FocusControl = HireDate		
end		
<> Object Label10: TLabel		
Left = 123		
Top = 116		
Width = 29		
Height = 13		
Caption = 'Salary'		
FocusControl = Salary		
end		
<> Object Label11: TLabel Left = 18		
Top = 18		
Width = 53		
Height = 16		
Alignment = taRightJustify		
Caption = 'Emp #: '		
Font.Charset = ANSI_CHARSET		
Font.Color = clBlack		
Font.Height = -13		
Font.Name = 'MS Sans Serif'		
Font.Style = [fsBold]		
ParentFont = False		
end		
<> Object DBText1: TDBText		
Left = 73		
Top = 18		
Width = 68		
Height = 16		
DataField = 'EmpNo'		
DataSource = EmpData		
Font.Charset = ANSI CHARSET		
Font.Color = clBlack		
Font.Height = -13		
Font.Name = 'MS Sans Serif'		
Font.Style = [fsBold]		
ParentFont = False		
end		
<> Object RecInd: TLabel		
Left = 346		
Top = 16		
Width = 3		
Height = 13		
end		
<> Object FirstName:		

<> Object Employees:	
TClientDataSet	
Aggregates = <>	
PacketRecords = 10	
Params = <>	
ProviderName =	
'EmpQueryProvider'	
RemoteServer = MidasConnection	
OnReconcileError =	
EmployeesReconcileError	
Left = 284	
Top = 12	
end	
<> Object MidasConnection:	
<b>TDCOMConnection</b>	
ServerGUID = '{53BC6562-5B3E-	
11D0-9FFC-00A0248E4B9A}'	
ServerName = 'Serv.EmpServer'	
Left = 387	
Top = 12	
end	
End	

procedure 89
الهلف لبرناهج
(Briefcase)
إعداد
علاء الدين اللباد
واتف ۷۱۳۵۷۵۶۹۰
program BriefcaseDemo;
uses
Forms,
BriefcaseMain in
'BriefcaseMain.pas' {Form1};
{\$R *.RES}
begin
Application.Initialize;
Application.CreateForm(TForm1,

Left = 304
Top = 88
Width = 105
Height = 25
Caption = '&Get Employees'
TabOrder = 5
OnClick = QueryButtonClick
end
<> Object DBNavigator1:
TDBNavigator
Left = 134
Top = 14 Width = 138
Height = 25
DataSource = EmpData
VisibleButtons = [nbFirst,
nbPrior, nbNext, nbLast, nbPost,
nbCancel]
TabOrder = 6
end
<> Object UpdateButton:
TButton
Left = 344
Top = 135
Width = 105
Height = 25
Caption = '&Update Employees'
TabOrder = 7
OnClick = UpdateButtonClick
end
<> Object UndoButton:
TButton
Left = 224
Top = 134
Width = 105
Height = 25
Caption = 'U&ndo Last Change'
TabOrder = 8
OnClick = UndoButtonClick
end
<> Object EmpData:
TDataSource
DataSet = Employees
OnDataChange =
<b>EmpDataDataChange</b>
Left = 331
Top = 36
end
V114

var CanClose: Boolean);	Form1);
<> procedure	Application.Run;
UpdateButtonClick(Sender:	end.
TObject);	
<> procedure	\\\\\\\\\\
ConnectionIndClick(Sender:	unit BriefcaseMain;
TObject);	
<> procedure	{ This program demonstrates how to
RefreshButtonClick(Sender:	do disconnected briefcase
TObject);	applications
private	with ADO. When the Connected
DataFileName: string;	checkbox is unchecked the
public	application is
<> procedure LoadData;	switched into offline mode. If the
<> procedure SaveData;	application is exited at that point
<> procedure UpdateData;	then the data is persisted to a file
end;	on disk (along with any edits to the
	data). When the application is
var	restarted it will load the persisted
Form1: TForm1;	data
,	if present, otherwise it will fetch the
implementation	data from the database. }
1	
{\$R *.dfm}	interface
const	uses
BaseFileName =	Windows, Messages, SysUtils,
'EMPLOYEE.ADTG';	Classes, Graphics, Controls, Forms,
	Dialogs,
procedure TForm1.LoadData;	StdCtrls, DB, ADODB, Grids,
begin	DBGrids, ExtCtrls;
DataFileName :=	
ExtractFilePath(Paramstr(0))+Base	type
FileName;	TForm1 = class(TForm)
{ If a persisted datafile exists,	<b>Employees: TADODataSet;</b>
assume we exited in a disconnected	EmpSource: TDataSource;
(offline) state and load the data	DBGrid1: TDBGrid;
from the file. }	Connection: TADOConnection;
if FileExists(DataFileName) then	Panel1: TPanel;
	ConnectionInd: TCheckBox;
Employees.LoadFromFile(DataFileN	UpdateButton: TButton;
ame)	RefreshButton: TButton;
else	SaveButton: TButton;
begin	<> procedure
{ Otherwise establish the	Form1Create(Sender: TObject);
connection and get data from the	<> procedure
database }	SaveButtonClick(Sender: TObject);
ConnectionInd.Checked := True;	<> procedure
Employees.Open;	Form1CloseQuery(Sender: TObject;

try	end;	
{ When closing, update the	end;	
database if connected or save it to	ciru,	
disk if not }	procedure TForm1.UpdateData;	
if Connection.Connected then	begin	
UpdateData else	{ Connect to the database and send	
SaveData;	the pending updates }	
except	ConnectionInd.Checked := True;	
on E: Exception do	Employees.UpdateBatch;	
begin	DeleteFile(DataFileName);	
begin	end;	
Application.HandleException(Self);	chu,	
CanClose := MessageDlg('Data	procedure TForm1.SaveData;	
not saved/updated, exit anyway?',	begin	
mtConfirmation,	{ Persist the data to disk }	
mbYesNoCancel, 0) = mrYes;	1 ci sist the data to disk ;	
end;	Employees.SaveToFile(DataFileNam	
end;	e, pfADTG);	
end;	end;	
ciu,	enu,	
procedure	procedure	
TForm1.ConnectionIndClick(Sender	TForm1.Form1Create(Sender:	
: TObject);	TObject);	
begin	begin	
{ Toggle the connection's state }	Connection.ConnectionString :=	
if ConnectionInd.Checked then	'FILE NAME=' + DataLinkDir +	
begin	'\DBDEMOS.UDL';	
Connection.Open;	LoadData;	
Employees.Connection :=	end;	
Connection;	chu,	
end else	procedure	
begin	TForm1.SaveButtonClick(Sender:	
{ Note here you must clear the	TObject);	
connection property of the dataset	begin	
before	SaveData;	
closing the connection.	end;	
Otherwise the dataset will close with	cau,	
the	procedure	
connection. }	TForm1.UpdateButtonClick(Sender:	
Employees.Connection := nil;	TObject);	
Connection.Close;	begin	
end;	UpdateData;	
end;	end;	
	· · · · · · · · · · · · · · · · · · ·	
procedure	procedure	
TForm1.RefreshButtonClick(Sender	TForm1.Form1CloseQuery(Sender:	
: TObject);	TObject; var CanClose: Boolean);	
begin	begin	
{ Close and reopen the dataset to	if Employees. Active then	
( Store and reopen the dataset to	ii Limpio jeesa tente men	

4925000#VI واتة	
object Form1: TForm1	
Left = 217	
Top = 143	
AutoScroll = False	
Caption = 'Briefcase Demo'	
ClientHeight = 332	
ClientWidth = 571	
Color = clBtnFace	
ParentFont = True	
OldCreateOrder = False	
Position = poScreenCenter	
OnCloseQuery =	
Form1CloseQuery	
OnCreate = Form1Create	
PixelsPerInch = 96	
TextHeight = 13	
object DBGrid1: TDBGrid	
Left = 0	
Top = 38	
Width = 571	
Height = 294	
Align = alClient	
DataSource = EmpSource	
TabOrder = 0	
TitleFont.Charset =	
DEFAULT_CHARSET	
TitleFont.Color = clWindowText	
TitleFont.Height = -11	
TitleFont.Name = 'MS Sans Serif'	
TitleFont.Style = []	
end	
object Panel1: TPanel	
Left = 0	
Top = 0	
Width = 571	
Height = 38	
Align = alTop	
BevelOuter = bvNone	
TabOrder = 1	
object ConnectionInd:	
TCheckBox	
Left = 58	
Top = 11	
Width = 79	
Height = 17	

refresh the data. Note that in this	
demo	
there is no checking for pending	
updates so they are lost if you click	
the refresh data button before	
clicking the Update database button.	
}	
ConnectionInd.Checked := True;	
Employees.Close;	
Employees.CommandType :=	
cmdTable;	
Employees.CommandText :=	
'Employee';	
Employees.Open;	
end;	
end.	
111111111111111111111111111111111111111	

ObjectS	
الهلف النصي لبرناهج	
(Briefcase)	
إعداد	
علاء الدين اللباد	

<b>TADOConnection</b>
ConnectionString =
'FILE NAME=C:\Program
Files\Common Files\System\OLE
DB\Data Links' +
'\DBDemos.UDL'
LoginPrompt = False
Mode = cmShareDenyNone
Provider =
'C:\Program Files\Common
Files\System\OLE DB\Data
Links\DBDemos.U' +
'DL'
Left = 13
Top = 3
end
End

Caption = 'Connected'
TabOrder = 0
OnClick = ConnectionIndClick
end
object UpdateButton: TButton
Left = 153
Top = 6
Width = 84
Height = 25
Caption = 'Update Server'
TabOrder = 1
OnClick = UpdateButtonClick
end
object RefreshButton: TButton
Left = 250
Top = 6
Width = 87
Height = 25
Caption = 'Refresh Data'
TabOrder = 2
OnClick = RefreshButtonClick
end
object SaveButton: TButton
Left = 352
Top = 6
Width = 75
Height = 25
Caption = 'Save to disk'
TabOrder = 3
OnClick = SaveButtonClick
end
end
object Employees: TADODataSet
Connection = Connection
CursorType = ctStatic
LockType = ltBatchOptimistic
CommandText = 'employee'
CommandType = cmdTable
Parameters = <>
Left = 12
Top = 89
end
object EmpSource: TDataSource
-
DataSet = Employees
Left = 69
Top = 97
end
object Connection:

علاء الدين اللباد	
واتف ۷۹٤٤٥٧٥٣٧١	
program Filter;	
uses	
Forms,	
DM in 'DM.pas' {DM1},	
CustView in 'CustView.pas'	
{fmCustView},	
About in 'About.pas' {fmAboutBox},	
Filter1 in 'Filter1.pas'	
{fmFilterFrm};	
{\$R *.RES}	
begin	
Application.Initialize;	
,	
Application.CreateForm(TfmCustVi	
ew, fmCustView);	
Application.CreateForm(TDM1, DM1);	
DMI),	
Application.CreateForm(TfmFilterF	
rm, fmFilterFrm);	
Application.Run;	
end.	
\\\\	
Unit DM;	
2	
interface	
HEGE	
Windows, Messages, SysUtils,	
Classes, Graphics, Controls, Forms,	
Dialogs,	
DB, DBTables;	
tyma	
type TDM1 = class(TDataModule)	
Customer: TTable;	
CustomerSource: TDataSource;	
SQLCustomer: TQuery;	
SQLOrders: TQuery;	
OrdersSource: TDataSource;	

Procedures 39	
الملف التنفيذي لبرنامج	
(Filter)	
عامدا	

TDM1.SQLOrdersFilterRecord(Dat	SQLOrdersOrderNo:	
aSet: TDataSet;	TFloatField;	
var Accept: Boolean);	SQLOrdersCustno: TFloatField;	
begin	SQLOrdersSaleDate:	
{ This is only called if the Filtered	TDateTimeField;	
property is True, set	SQLOrdersShipDate:	
dynamically by the CheckBox on	TDateTimeField;	
the CustView form. }	SQLOrdersEmpNo:	
Accept :=	TIntegerField;	
SQLOrdersAmountPaid.Value >=	SQLOrdersAmountPaid:	
OrdersFilterAmount;	TCurrencyField;	
end;	<> procedure	
	DM1Create(Sender: TObject);	
end.	<> procedure	
	SQLOrdersFilterRecord(DataSet:	
\\\\	TDataSet;	
	var Accept: Boolean);	
unit About;	public	
	{ The variable below will be	
interface	accessible to to CustView (because it	
	is	
uses	public and this unit is in its uses).	
Windows, Messages, SysUtils,	It is used in	
Classes, Graphics, Controls, Forms,	SQLOrdersFilterRecord to set	
Dialogs,	the Filter amount for the Orders	
StdCtrls, ExtCtrls;	Query. }	
	OrdersFilterAmount: Extended;	
type	end;	
TfmAboutBox = class(TForm)		
Panel1: TPanel;	var	
ProgramIcon: TImage;	DM1: TDM1;	
ProductName: TLabel;		
Version: TLabel;	implementation	
Memo1: TMemo;		
OKButton: TButton;	{\$R *.dfm}	
end;		
	<> procedure	
var	TDM1.DM1Create(Sender:	
fmAboutBox: TfmAboutBox;	TObject);	
	begin	
implementation	try	
	Screen.Cursor := crHourGlass;	
{\$R *.dfm}	SQLCustomer.Open;	
	finally	
end.	Screen.Cursor := crDefault;	
	end;	
111111111111111111111111111111111111111	end;	
unit About;		
·	<> procedure	

About1: TMenuItem;
DBNavigator1: TDBNavigator;
Label2: TLabel;
rgDataSet: TRadioGroup;
SpeedButton1: TSpeedButton;
DBGrid1: TDBGrid;
GroupBox1: TGroupBox;
cbFilterOrders: TCheckBox;
Label1: TLabel;
Edit1: TEdit;
<> procedure
rgDataSetClick(Sender: TObject);
<> procedure
SpeedButton1Click(Sender:
TObject);
<> procedure
cbFilterOrdersClick(Sender:
TObject);
<> procedure
About1Click(Sender: TObject);
<> procedure
Edit1Change(Sender: TObject);
<> procedure
DBGrid1Enter(Sender: TObject);
<> procedure
DBGrid2Enter(Sender: TObject);
end;
var
fmCustView: TfmCustView;
Ź
implementation
•
uses DM, Filter1, About;
{\$R *.dfm}
(wit turn)
{ Change the Dataset for the
Customer from Query to Table,
or Table to Query. If a filter is
current, set that filter
to the just-changed Dataset. }
<> procedure
TfmCustView.rgDataSetClick(Sende
r: TObject);
var
st: string;
begin
with DM1, CustomerSource do
with Divii, Customer Source do

interfa	
IIICIIa	
uses	
	lows, Messages, SysUtils,
	, 8 , ,
	s, Graphics, Controls, Forms,
Dialog	
StaC	trls, ExtCtrls;
type	(D) 1 (TE)
	AboutBox = class(TForm)
	el1: TPanel;
	gramIcon: TImage;
	ductName: TLabel;
	sion: TLabel;
Men	no1: TMemo;
OKI	Button: TButton;
end;	
var	
fmAb	outBox: TfmAboutBox;
	·
implen	nentation
{\$R *.0	dfm}
	,
end.	
\\\\\\	\\\\\\\\
unit C	ustView;
unit C	ust view,
interfa	
IIICIIa	
uses	
	owe Mossogos Svelitile
	lows, Messages, SysUtils, s, Graphics, Controls, Forms,
Dialog	<u>s,</u> trls, DBCtrls, Grids, DBGrids,
	trls, StdCtrls, Mask, Buttons,
Menu	is;
type	
	CustView = class(TForm)
	· · · · · · · · · · · · · · · · · · ·
Stat	usBar1: TStatusBar;
Stat	· · · · · · · · · · · · · · · · · · ·
Stat DB(	usBar1: TStatusBar;

Edit1Change(nil). (To make sure	begin	
Edit1Change(nil); { To make sure		
the value is set. }	{ Is the other Dataset Filtered?	
end;	Get its filter. }	
	if Dataset.Filtered then	
<> procedure	st := Dataset.Filter;	
TfmCustView.About1Click(Sender:	case rgDataset.ItemIndex of	
TObject);	0: if Dataset <> SQLCustomer	
begin	then	
with TFMAboutBox.Create(nil) do	Dataset := SQLCustomer;	
try	1: if CustomerSource.Dataset <>	
ShowModal;	<b>Customer then</b>	
finally	Dataset := Customer;	
Free;	end;	
end;		
end;	{ Set the Filter of the current	
	Dataset. }	
<> procedure	if st <> " then	
TfmCustView.Edit1Change(Sender:	begin	
TObject);	Dataset.Filter := st;	
begin	Dataset.Filtered := True;	
{ We must test of Edit1.Text not	end;	
being blank, or an exception will	end;	
be raised when the user clears the	end;	
contents of the Edit. }		
if (cbFilterOrders.checked) and	<> procedure	
(Edit1.Text <> '') then	TfmCustView.SpeedButton1Click(S	
try	ender: TObject);	
{ Attempt to set the variable on	begin	
DM1 that holds the filter amount. }	fmFilterFrm.Show;	
DM1.OrdersFilterAmount :=	end;	
StrToFloat(fmCustView.Edit1.Text)	ciu;	
:	<> procedure	
	TfmCustView.cbFilterOrdersClick(	
{ Refreshing SQLOrders will	Sender: TObject);	
cause DM1.SQLOrdersFilterRecord	begin	
to be called	{ DM1.SQLOrders.Filtered will be	
once for each orders record for	true if the box is checked, false	
the current Customer. }	otherwise. }	
DM1.SQLOrders.Refresh;	DM1.SQLOrders.Filtered :=	
	cbFilterOrders.Checked;	
except	The state of the s	
on EConvertError do	{ If the number changed while the	
{ This happens before	box is unchecked,	
DM1.SQLOrders.refresh if the user	DM1.OrdersFilterAmount	
typed in a	won't know about it. The	
non-numeric value, thus	Edit1Change <> procedure	
preventing it from trying to use an	will set the value and	
invalid	apply the filter.}	
value to filter itself, which	if cbFilterOrders.Checked then	
variate to inited leading winter	ii cornici Oruci S. Checkeu theli	

Labelle TLabels	would using an averaging area for
Label3: TLabel;	would raise an exception once for
Memo1: TMemo;	each record displayed.}
GroupBox1: TGroupBox;	raise
cbCaseSensitive: TCheckBox;	Exception.Create('Threshold
cbNoPartialCompare:	Amount must be a number')
TCheckBox;	end
ComboBox1: TComboBox;	end;
Label4: TLabel;	
BtnApplyFilter: TButton;	{ Set the navigator to the Customer
BtnClear: TButton;	Datasource. }
BtnClose: TButton;	<> procedure
<> procedure	TfmCustView.DBGrid1Enter(Sende
AddFieldName(Sender: TObject);	r: TObject);
<> procedure	begin
ListBox2DblClick(Sender: TObject);	DBNavigator1.DataSource :=
<> procedure	DBGrid1.DataSource;
ApplyFilter(Sender: TObject);	end;
<> procedure	
FormCreate(Sender: TObject);	{ Set the navigator to the Orders
<> procedure	Datasource. }
Memo1Change(Sender: TObject);	<> procedure
<> procedure	TfmCustView.DBGrid2Enter(Sende
cbCaseSensitiveClick(Sender:	r: TObject);
TObject);	begin
<> procedure	DBNavigator1.DataSource :=
cbNoPartialCompareClick(Sender:	DBGrid2.DataSource;
TObject);	end;
<> procedure	
SBtnClearClick(Sender: TObject);	end.
<> procedure	
ComboBox1Change(Sender:	
TObject);	
<> procedure	
SBtnCloseClick(Sender: TObject);	unit Filter1;
end;	
	interface
var	
fmFilterFrm: TfmFilterFrm;	uses
	Windows, Messages, SysUtils,
implementation	Classes, Graphics, Controls, Forms,
	Dialogs,
uses DM, CustView;	StdCtrls, Buttons, DB;
{\$R *.dfm}	type
	TfmFilterFrm = class(TForm)
{ Adds current listbox field name to	Label1: TLabel;
memo. }	Label: TLabel;
<> procedure	ListBox1: TListBox;
TfmFilterFrm.AddFieldName(Sende	ListBox2: TListBox;
	LIBUDAN I LIBUDA,

TOL:	TOL: A
TObject);	r: TObject);
var	begin
I: Integer;	if Memo1.Text <> " then
begin	Memo1.Text := Memo1.Text + ' ';
for I := 0 to	Memo1.Text := Memo1.Text +
DM1.CustomerSource.Dataset.Field	ListBox1.Items[ListBox1.ItemIndex]
Count - 1 do	;
	end;
ListBox1.Items.Add(DM1.Customer.	
Fields[I].FieldName);	{ Adds current Filter operator to
	memo. }
{ Add date dependent Query	<> procedure
ComboBox1. }	TfmFilterFrm.ListBox2DblClick(Se
,	nder: TObject);
ComboBox1.Items.Add('LastInvoice	begin
Date >= ''' +	if Memo1.Text <> " then
DateToStr(EncodeDate(1994, 09,	Memo1.Text := Memo1.Text + ' '+
30)) + ''');	ListBox2.Items[ListBox2.ItemIndex]
ComboBox1.Items.Add('Country =	Listbox2.items[Listbox2.itemindex]
"US" and LastInvoiceDate > "" +	onds.
DateToStr(EncodeDate(1994, 06,	end;
30)) + '''');	<> procedure
end;	TfmFilterFrm.ApplyFilter(Sender:
( 0 ) 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	TObject);
{ Since the Filter property is a	begin
TStrings and the Memo field	with DM1.CustomerSource.Dataset
is a TMemo, convert the Memo's	do
wrapped text to a string,	begin
which is then used when the user	if ComboBox1.Text <> " then
presses Apply. }	begin
<> procedure	Filter := ComboBox1.Text;
TfmFilterFrm.Memo1Change(Sende	Filtered := True;
r: TObject);	fmCustView.Caption :=
var	'Customers - Filtered';
I: Integer;	end
begin	else begin
ComboBox1.Text :=	Filter := ";
Memo1.Lines[0];	Filtered := False;
for I := 1 to Memo1.Lines.Count - 1	fmCustView.Caption :=
do	'Customers - Unfiltered'
ComboBox1.Text :=	end;
ComboBox1.Text + ' ' +	end;
Memo1.Lines[I];	end;
end;	enu,
	( Donulate the T tetD == 1 1/4
{ Set the Customer's Dataset Case	{ Populate the ListBox1 with
Sensitive Filter Option. }	available fields from the Customer
-	Dataset. }
<pre>&lt;&gt; procedure TfmFiltonFrm ab CaseS anaitive Click(</pre>	<> procedure
TfmFilterFrm.cbCaseSensitiveClick(	TfmFilterFrm.FormCreate(Sender:

ender: TObject);
begin
Memo1.Lines.Clear;
Memo1.Lines.Add(ComboBox1.Text);
end;
{ Close the Filter Form. }
<> procedure
TfmFilterFrm.SBtnCloseClick(Send
er: TObject);
begin
Close;
end;
end.

<> Object\$50
الملف النصي لبرنامج
(Filter)
عداد
علاء الدين اللباد
واتف ۷۱۳۵۷۵۶۹۰
<> Object fmCustView:
TfmCustView
Left = 227
Top = 109
BorderStyle = bsSingle
Caption = 'Customers - Unfiltered'
ClientHeight = 362
ClientWidth = 493
Color = clBtnFace
ParentFont = True
Menu = MainMenu1
OldCreateOrder = True
PixelsPerInch = 96
TextHeight = 13
<> Object Label3: TLabel

Sender: TObject);
begin
with DM1.CustomerSource.Dataset
do
if cbCaseSensitive.checked then
FilterOptions := FilterOptions -
[foCaseInSensitive]
•
else
FilterOptions := FilterOptions +
[foCaseInsensitive];
end;
{ Set the Customer Partial Compare
Filter Option. }
<> procedure
TfmFilterFrm.cbNoPartialCompare
Click(Sender: TObject);
begin
with DM1.CustomerSource.Dataset
do
if cbNoPartialCompare.checked
then
FilterOptions := FilterOptions +
[foNoPartialCompare]
else
FilterOptions := FilterOptions -
[foNoPartialCompare];
end;
{ Add User-Entered filters into list
box at runtime. }
<> procedure
TfmFilterFrm.SBtnClearClick(Send
er: TObject);
ž / ·
var
st: string;
begin
Memo1.Lines.Clear;
st := ComboBox1.Text;
ComboBox1.Text := ";
if ComboBox1.Items.IndexOf(st) =
-1 then
ComboBox1.Items.Add(st);
end;
,
{ Reset the Memo field when the
Filter ComboBox changes. }
<> procedure
TfmFilterFrm.ComboBox1Change(S

Top = 208
Width = 350
Height = 129
DataSource = DM1.OrdersSource
TabOrder = 1
TitleFont.Charset =
DEFAULT_CHARSET
TitleFont.Color = clWindowText
TitleFont.Height = -11
TitleFont.Name = 'MS Sans Serif'
TitleFont.Style = []
OnEnter = DBGrid2Enter
Columns = <
item
Expanded = False
FieldName = 'OrderNo'
Visible = True
end
item
Expanded = False
FieldName = 'Custno'
Title.Caption = 'CustNo'
Width = 45
Visible = True
end
item
Expanded = False
FieldName = 'AmountPaid'
Title.Caption = 'Amount Paid'
Width = 122
Visible = True
end
item
Expanded = False
FieldName = 'ShipDate'
Title.Caption = 'Ship Date'
Width = 83
Visible = True
end>
end
<> Object DBNavigator1: TDBNavigator
Left = 132
Top = 4 Width = 88
Height = 25 DataSource =
DataSource -
DM1.CustomerSource

Left = 12
Top = 192
Width = 93
Height = 16
Caption = 'Orders Query'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
<> Object Label2: TLabel
Left = 8
Top = 36
<b>Width</b> = 74
Height = 16
Caption = 'Customers'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
<pre>&lt;&gt; Object SpeedButton1: TSpeedPutton</pre>
TSpeedButton
Left = 9
Top = 128
Width = 96
Height = 25
Caption = 'Filter &Customers'
NumGlyphs = 2
OnClick = SpeedButton1Click
end
<> Object StatusBar1:
TStatusBar
Left = 0
Top = 343
Width = 493
Height = 19
Panels = <>
end
<> Object DBGrid2:
TDBGrid
Left = 132

Expanded = False
FieldName = 'Country'
Width = 64
Visible = True
end
item
Expanded = False
FieldName = 'LastInvoiceDate'
Title.Caption = 'Last Invoice'
<b>Width</b> = 64
Visible = True
end
item
Expanded = False
FieldName = 'Contact'
<b>Width</b> = 64
Visible = True
end
item
Expanded = False
FieldName = 'City'
Width = 64
Visible = True
end
item
Expanded = False
FieldName = 'State'
Visible = True
end>
end
<> Object GroupBox1:
TGroupBox
Left = 7
Top = 224
Width = 121
Height = 81
Caption = 'Orders Filter'
TabOrder = 5
<> Object Label1: TLabel
Left = 4
Top = 52
Width = 60
Height = 13
Caption = 'Amount & Paid'
FocusControl = Edit1
end
<> Object cbFilterOrders:
TCheckBox

VisibleButtons = [nbFirst,
nbPrior, nbNext, nbLast]
TabOrder = 3
end
<> Object rgDataSet:
TRadioGroup
Left = 8
Top = 60
Width = 97
Height = 61
Caption = 'Dataset'
ItemIndex = 1
Items.Strings = (
'&Query-Based'
'&Table-Based')
TabOrder = 2
OnClick = rgDataSetClick
end
<> Object DBGrid1:
TDBGrid
Left = 132
Top = 33
Width = 350
Height = 163
DataSource =
DM1.CustomerSource
TabOrder = 0
TitleFont.Charset =
DEFAULT_CHARSET
TitleFont.Color = clWindowText
TitleFont.Height = -11
TitleFont.Name = 'MS Sans Serif'
TitleFont.Style = []
OnEnter = DBGrid1Enter
Columns = <
item
Expanded = False
FieldName = 'CustNo'
Width = 43
Visible = True
end
item
Expanded = False
FieldName = 'Company'
Width = 64
Visible = True
end
item

<> Object CustomerSource:
TDataSource Data Source
DataSet = Customer
Left = 111
Top = 33
end
<> Object SQLCustomer:
TQuery
DatabaseName = 'DBDEMOS'
SQL.Strings = (
'SELECT * FROM
"CUSTOMER.DB" ')
Left = 26
Top = 62
end
<> Object SQLOrders:
TQuery
Active = True
DatabaseName = 'DBDEMOS'
OnFilterRecord =
SQLOrdersFilterRecord
DataSource = CustomerSource
SQL.Strings = (
'Select * From Orders Where
Custno = :CustNo ')
Left = 196
Top = 32 ParamData = <
item
DataType = ftFloat
Name = 'CustNo'
ParamType = ptUnknown
Size = 8
end>
<> Object
SQLOrdersOrderNo: TFloatField
DisplayWidth = 10 FieldName = 'OrderNo'
end end
<> Object
SQLOrdersCustno: TFloatField
DisplayWidth = 8  FieldName = !Custne!
FieldName = 'Custno'
end Chicat
<> Object
SQLOrdersAmountPaid:
TCurrencyField DisplayWidth = 13
DisplayWidth = 13

Left = 8
Top = 20
Width = 109
Height = 17
Caption = '&Filter'
TabOrder = 0
OnClick = cbFilterOrdersClick
end
<> Object Edit1: TEdit
Left = 68
Top = 48
Width = 49
Height = 21
TabOrder = 1
Text = '1000'
OnChange = Edit1Change
end
end
<> Object MainMenu1:
TMainMenu
Left = 104
Top = 5
<> Object About1:
TMenuItem
Caption = '&About'
OnClick = About1Click
end
end
End
Ditt.
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
<> Object DM1: TDM1
OldCreateOrder = True
OnCreate = DM1Create
Left = 175
Top = 163
Height = 223
Width = 451
<> Object Customer: TTable
Active = True
DatabaseName = 'DBDEMOS'
Filtered = True
IndexFieldNames = 'Company'
TableName = 'CUSTOMER.DB'
Left = 27
Top = 9
10p /

Left = 8  Top = 128  Width = 27  Height = 13  Caption = '&Fields'  FocusControl = ListBox1 end  <> Object Label2: TLabel  Left = 8  Top = 44  Width = 44  Height = 13  Caption = '&Condition' end  <> Object Label3: TLabel  Left = 161  Top = 128  Width = 46  Height = 13  Caption = '&Operators' end  <> Object Label4: TLabel  Left = 8  Top = 8  Width = 16  Height = 13  Caption = '&List'  FocusControl = ComboBox1 end  <> Object ListBox1: TListBox  Left = 8  Top = 144  Width = 143  Height = 73  Hint = 'Doubleclick to add'  TabStop = False  ItemHeight = 13  ParentShowHint = False  ShowHint = True  Sorted = True  TabOrder = 2  OnDblClick = AddFieldName end  <> Object ListBox2: TListBox  Left = 161  Top = 144	
Width = 27 Height = 13 Caption = '&Fields' FocusControl = ListBox1 end <> Object Label2: TLabel Left = 8 Top = 44 Width = 44 Height = 13 Caption = '&Condition' end <> Object Label3: TLabel Left = 161 Top = 128 Width = 46 Height = 13 Caption = '&Operators' end <> Object Label4: TLabel Left = 8 Top = 8 Width = 16 Height = 13 Caption = '&List' FocusControl = ComboBox1 end <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	Left = 8
Height = 13 Caption = '&Fields' FocusControl = ListBox1 end <> Object Label2: TLabel Left = 8 Top = 44 Width = 44 Height = 13 Caption = '&Condition' end <> Object Label3: TLabel Left = 161 Top = 128 Width = 46 Height = 13 Caption = '&Operators' end <> Object Label4: TLabel Left = 8 Top = 8 Width = 16 Height = 13 Caption = '&List' FocusControl = ComboBox1 end <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	
Caption = '&Fields' FocusControl = ListBox1 end <> Object Label2: TLabel Left = 8 Top = 44 Width = 44 Height = 13 Caption = '&Condition' end <> Object Label3: TLabel Left = 161 Top = 128 Width = 46 Height = 13 Caption = '&Operators' end <> Object Label4: TLabel Left = 8 Top = 8 Width = 16 Height = 13 Caption = '&List' FocusControl = ComboBox1 end <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	
FocusControl = ListBox1 end  <> Object Label2: TLabel Left = 8 Top = 44 Width = 44 Height = 13 Caption = '&Condition' end  <> Object Label3: TLabel Left = 161 Top = 128 Width = 46 Height = 13 Caption = '&Operators' end  <> Object Label4: TLabel Left = 8 Top = 8 Width = 16 Height = 13 Caption = '&List' FocusControl = ComboBox1 end  <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end  <> Object ListBox2: TListBox Left = 161	
end > Object Label2: TLabel Left = 8 Top = 44 Width = 44 Height = 13 Caption = '&Condition' end> Object Label3: TLabel Left = 161 Top = 128 Width = 46 Height = 13 Caption = '&Operators' end> Object Label4: TLabel Left = 8 Top = 8 Width = 16 Height = 13 Caption = '&List' FocusControl = ComboBox1 end> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end> Object ListBox2: TListBox Left = 161	
C> Object Label2: TLabel Left = 8 Top = 44 Width = 44 Height = 13 Caption = '&Condition' end C> Object Label3: TLabel Left = 161 Top = 128 Width = 46 Height = 13 Caption = '&Operators' end C> Object Label4: TLabel Left = 8 Top = 8 Width = 16 Height = 13 Caption = '&List' FocusControl = ComboBox1 end C> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end C> Object ListBox2: TListBox Left = 161	FocusControl = ListBox1
Left = 8 Top = 44 Width = 44 Height = 13 Caption = '&Condition' end <> Object Label3: TLabel Left = 161 Top = 128 Width = 46 Height = 13 Caption = '&Operators' end <> Object Label4: TLabel Left = 8 Top = 8 Width = 16 Height = 13 Caption = '&List' FocusControl = ComboBox1 end <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	end
Top = 44 Width = 44 Height = 13 Caption = '&Condition' end <> Object Label3: TLabel Left = 161 Top = 128 Width = 46 Height = 13 Caption = '&Operators' end <> Object Label4: TLabel Left = 8 Top = 8 Width = 16 Height = 13 Caption = '&List' FocusControl = ComboBox1 end <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	<> Object Label2: TLabel
Width = 44 Height = 13 Caption = '&Condition' end <> Object Label3: TLabel Left = 161 Top = 128 Width = 46 Height = 13 Caption = '&Operators' end <> Object Label4: TLabel Left = 8 Top = 8 Width = 16 Height = 13 Caption = '&List' FocusControl = ComboBox1 end <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	Left = 8
Height = 13 Caption = '&Condition' end <> Object Label3: TLabel Left = 161 Top = 128 Width = 46 Height = 13 Caption = '&Operators' end <> Object Label4: TLabel Left = 8 Top = 8 Width = 16 Height = 13 Caption = '&List' FocusControl = ComboBox1 end <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	Top = 44
Caption = '&Condition' end  <> Object Label3: TLabel Left = 161 Top = 128 Width = 46 Height = 13 Caption = '&Operators' end  <> Object Label4: TLabel Left = 8 Top = 8 Width = 16 Height = 13 Caption = '&List' FocusControl = ComboBox1 end  <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end  <> Object ListBox2: TListBox Left = 161	Width = 44
end  <> Object Label3: TLabel  Left = 161  Top = 128  Width = 46  Height = 13  Caption = '&Operators' end  <> Object Label4: TLabel  Left = 8  Top = 8  Width = 16  Height = 13  Caption = '&List'  FocusControl = ComboBox1 end  <> Object ListBox1:  TListBox  Left = 8  Top = 144  Width = 143  Height = 73  Hint = 'Doubleclick to add'  TabStop = False  ItemHeight = 13  ParentShowHint = False  ShowHint = True  Sorted = True  TabOrder = 2  OnDblClick = AddFieldName end  <> Object ListBox2:  TListBox  Left = 161	Height = 13
<pre>&lt;&gt; Object Label3: TLabel   Left = 161   Top = 128   Width = 46   Height = 13   Caption = '&amp;Operators'   end   &lt;&gt; Object Label4: TLabel   Left = 8   Top = 8   Width = 16   Height = 13   Caption = '&amp;List'   FocusControl = ComboBox1   end   &lt;&gt; Object ListBox1:   TListBox   Left = 8   Top = 144   Width = 143   Height = 73   Hint = 'Doubleclick to add'   TabStop = False   ItemHeight = 13   ParentShowHint = False   ShowHint = True   Sorted = True   TabOrder = 2   OnDblClick = AddFieldName   end   &lt;&gt; Object ListBox2:   TListBox   Left = 161</pre>	Caption = '&Condition'
Left = 161 Top = 128 Width = 46 Height = 13 Caption = '&Operators' end <> Object Label4: TLabel Left = 8 Top = 8 Width = 16 Height = 13 Caption = '&List' FocusControl = ComboBox1 end <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	end
Left = 161 Top = 128 Width = 46 Height = 13 Caption = '&Operators' end <> Object Label4: TLabel Left = 8 Top = 8 Width = 16 Height = 13 Caption = '&List' FocusControl = ComboBox1 end <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	<> Object Label3: TLabel
Width = 46 Height = 13 Caption = '&Operators' end <> Object Label4: TLabel Left = 8 Top = 8 Width = 16 Height = 13 Caption = '&List' FocusControl = ComboBox1 end <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	
Width = 46 Height = 13 Caption = '&Operators' end <> Object Label4: TLabel Left = 8 Top = 8 Width = 16 Height = 13 Caption = '&List' FocusControl = ComboBox1 end <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	Top = 128
Height = 13 Caption = '&Operators' end <> Object Label4: TLabel Left = 8 Top = 8 Width = 16 Height = 13 Caption = '&List' FocusControl = ComboBox1 end <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	
Caption = '&Operators' end  <> Object Label4: TLabel Left = 8 Top = 8 Width = 16 Height = 13 Caption = '&List' FocusControl = ComboBox1 end <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	
end <> Object Label4: TLabel Left = 8 Top = 8 Width = 16 Height = 13 Caption = '&List' FocusControl = ComboBox1 end <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	
<pre>&lt;&gt; Object Label4: TLabel   Left = 8   Top = 8   Width = 16   Height = 13   Caption = '&amp;List'   FocusControl = ComboBox1   end   &lt;&gt; Object ListBox1:   TListBox   Left = 8   Top = 144   Width = 143   Height = 73   Hint = 'Doubleclick to add'   TabStop = False   ItemHeight = 13   ParentShowHint = False   ShowHint = True   Sorted = True   TabOrder = 2   OnDblClick = AddFieldName   end   &lt;&gt; Object ListBox2:   TListBox   Left = 161</pre>	
Left = 8  Top = 8  Width = 16  Height = 13  Caption = '&List'  FocusControl = ComboBox1 end  <> Object ListBox1:  TListBox  Left = 8  Top = 144  Width = 143  Height = 73  Hint = 'Doubleclick to add'  TabStop = False  ItemHeight = 13  ParentShowHint = False  ShowHint = True  Sorted = True  TabOrder = 2  OnDblClick = AddFieldName end  <> Object ListBox2:  TListBox  Left = 161	
Top = 8  Width = 16  Height = 13  Caption = '&List'  FocusControl = ComboBox1  end  <> Object ListBox1:  TListBox  Left = 8  Top = 144  Width = 143  Height = 73  Hint = 'Doubleclick to add'  TabStop = False  ItemHeight = 13  ParentShowHint = False  ShowHint = True  Sorted = True  TabOrder = 2  OnDblClick = AddFieldName  end  <> Object ListBox2:  TListBox  Left = 161	
Width = 16 Height = 13 Caption = '&List' FocusControl = ComboBox1 end <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	
Height = 13 Caption = '&List' FocusControl = ComboBox1 end <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	-
Caption = '&List' FocusControl = ComboBox1 end <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	
FocusControl = ComboBox1 end <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	
end <> Object ListBox1: TListBox Left = 8 Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	•
<> Object ListBox1: TListBox  Left = 8  Top = 144  Width = 143  Height = 73  Hint = 'Doubleclick to add'  TabStop = False  ItemHeight = 13  ParentShowHint = False  ShowHint = True  Sorted = True  TabOrder = 2  OnDblClick = AddFieldName end  <> Object ListBox2: TListBox  Left = 161	
TListBox  Left = 8  Top = 144  Width = 143  Height = 73  Hint = 'Doubleclick to add'  TabStop = False  ItemHeight = 13  ParentShowHint = False  ShowHint = True  Sorted = True  TabOrder = 2  OnDblClick = AddFieldName  end  <> Object ListBox2:  TListBox  Left = 161	
Left = 8  Top = 144  Width = 143  Height = 73  Hint = 'Doubleclick to add'  TabStop = False  ItemHeight = 13  ParentShowHint = False  ShowHint = True  Sorted = True  TabOrder = 2  OnDblClick = AddFieldName end  <> Object ListBox2:  TListBox  Left = 161	
Top = 144 Width = 143 Height = 73 Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	
Width = 143  Height = 73  Hint = 'Doubleclick to add'  TabStop = False  ItemHeight = 13  ParentShowHint = False  ShowHint = True  Sorted = True  TabOrder = 2  OnDblClick = AddFieldName end  <> Object ListBox2:  TListBox  Left = 161	
Height = 73  Hint = 'Doubleclick to add'  TabStop = False  ItemHeight = 13  ParentShowHint = False  ShowHint = True  Sorted = True  TabOrder = 2  OnDblClick = AddFieldName end  <> Object ListBox2:  TListBox  Left = 161	
Hint = 'Doubleclick to add' TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	
TabStop = False ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	
ItemHeight = 13 ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	
ParentShowHint = False ShowHint = True Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	
ShowHint = True  Sorted = True  TabOrder = 2  OnDblClick = AddFieldName end  <> Object ListBox2: TListBox Left = 161	S
Sorted = True TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	
TabOrder = 2 OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	
OnDblClick = AddFieldName end <> Object ListBox2: TListBox Left = 161	
end <> Object ListBox2: TListBox Left = 161	
<> Object ListBox2: TListBox Left = 161	
TListBox Left = 161	
Left = 161	1
Top = 144	
	Top = 144

FieldName = 'AmountPaid'
end
<> Object
SQLOrdersSaleDate:
<b>TDateTimeField</b>
Alignment = taRightJustify
DisplayWidth = 10
FieldName = 'SaleDate'
end
<> Object
SQLOrdersShipDate:
TDateTimeField
Alignment = taRightJustify
DisplayWidth = 10
FieldName = 'ShipDate'
end
<> Object
SQLOrdersEmpNo: TIntegerField
DisplayWidth = 8
FieldName = 'EmpNo'
end
end
<> Object OrdersSource:
TDataSource TDataSource
DataSet = SQLOrders
Left = 272
Top = 32
end
End
Ellu
111111111111111111111111111111111111111
<> Object fmFilterFrm:
TfmFilterFrm
Left = 332
Top = 132
ActiveControl = Memo1
BorderStyle = bsSingle
Caption = 'Filter Condition'
ClientHeight = 298
ClientWidth = 305
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> Object Label1: TLabel

cbNoPartialCompare: TCheckBox
Left = 8
Top = 36
Width = 125
Height = 17
Hint = 'Disable partial string
comparisons (strings ending with
Caption = 'No Partial Compare'
TabOrder = 1
OnClick =
cbNoPartialCompareClick
end
end
<> Object ComboBox1:
TComboBox
Left = 8
Top = 24
Width = 289
Height = 21
Cursor = crArrow
ItemHeight = 13
TabOrder = 0
OnChange = ComboBox1Change
Items.Strings = (
'Company = "S*"
'Country <> "US""
'Country = ''US''')
end
<> Object BtnApplyFilter:
TButton
Left = 224
Top = 143
Width = 75
Height = 25
Caption = '&Apply'
TabOrder = 5
OnClick = ApplyFilter
end
<> Object BtnClear: TButton
Left = 224
Top = 173
Width = 75
Height = 25
Caption = 'Cl&ear'
TabOrder = 6
OnClick = SBtnClearClick
end
* == **

<b>Width</b> = 53
Height = 141
Hint = 'Double-click to add'
TabStop = False
ItemHeight = 13
Items.Strings = (
'>'
'<'
'= '
'>='
'<='
' <b>⇔</b> '
'AND'
'OR'
'('
CC
ParentShowHint = False
ShowHint = True
TabOrder = 3
OnDblClick = ListBox2DblClick
end
<> Object Memo1: TMemo
Left = 8
Top = 60
Width = 289
Height = 65
TabOrder = 1
OnChange = Memo1Change
end
<> Object GroupBox1:
TGroupBox
Left = 12
Top = 224
<b>Width = 139</b>
Height = 61
Caption = 'F&ilter Options'
TabOrder = 4
<> Object cbCaseSensitive:
TCheckBox
Left = 8
Top = 16
Width = 97
Height = 17
Caption = 'Case Sensitive'
TabOrder = 0
OnClick = cbCaseSensitiveClick
end
<> Object

<> Object ProductName:
TLabel
Left = 88
Top = 16
Width = 65
Height = 13
Caption = 'Filter Example'
IsControl = True
end
<> Object Version: TLabel
Left = 88
Top = 36
Width = 44
Height = 13
Caption = 'Version 1'
IsControl = True
end
<> Object Memo1: TMemo
Left = 8
Top = 72
Width = 265
Height = 181
Lines.Strings = (
'This example illustrates'
- The use of a Data Module
with a '
' Query linked through its
Datasource'
' property for displaying detail records.'
'- Switching Between Query- and Table-'
' Based datasets.'
'- Filtering Records
dynamically, by setting '
' the Filter property of a
Dataset.'
'- Filtering Records more
statically, '
by providing a value to a
variable hooked'
' to the OnFilterRecord event
· ·
"
'At runtime, the Data Module''s
only master '
'datasource can be connected to
either the '
etther the

<> Object BtnClose: TButton
Left = 224
Top = 256
Width = 75
Height = 25
Caption = 'Cl&ose'
TabOrder = 7
OnClick = SBtnCloseClick
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
<> Object fmAboutBox:
TfmAboutBox
Left = 442
Top = 133
BorderStyle = bsDialog
Caption = 'fmAboutBox'
ClientHeight = 321
ClientWidth = 299
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
PixelsPerInch = 96
TextHeight = 13
<> Object Panel1: TPanel
Left = 8
Top = 8
Width = 281
Height = 265
BevelInner = bvRaised
BevelOuter = bvLowered
Caption = 'Panel1'
TabOrder = 0
<> Object ProgramIcon:
TImage Left = 8
Top = 8
Width = 65
Height = 57
Picture.Data = {
0000}
Stretch = True
IsControl = True
end

الملفالتنفيذي لبرنامج	'Customer TQuery or the Customer TTable.'
(GdsDemo)	'This Dataset is linked to a
	single Datasource '
عامدا	'which serves as the link to
علاء الدين اللباد	another dataset,'
	'defined by a dynamic query of
الله المهره ١٩٤٤٥٧٥ عاده	the Orders table.'
program GdsDemo;	'Filtering is done on the
,	Customer records '
uses	'dynamically, with the user
Forms,	providing an'
GdsStd in 'GdsStd.pas'	'arbitrary expression. In
{GDSStdForm},	contrast, the Filter '
GdsData in 'GdsData.pas'	'button for the Orders table is
{StdDataForm},	done in a '
RecForm in 'RecForm.pas'	'more static, code-based
{RecViewForm},	fashion, with the'
GridForm in 'GridForm.pas'	'user simply providing a
{GridViewForm};	threshold for the '
	'Total Order Amount. An
{\$R *.RES}	exception is raised'
	'when user entries are
begin	inappropriate.')
Application.Initialize;	ReadOnly = True
	ScrollBars = ssVertical
Application.CreateForm(TGridView	TabOrder = 0
Form, GridViewForm);	end
And in the Constant of the William	end
Application.CreateForm(TRecView Form, RecViewForm);	<> Object OKButton:
Application.Run;	TButton
end.	Left = 111
enu.	Top = 284 Width = 77
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	Width = 77  Height = 27
	Caption = 'OK'
unit GdsStd;	Default = True
unit Gusseu,	ModalResult = 1
interface	TabOrder = 1
interrace	end
uses	End
SysUtils, Windows, Messages,	Ditt
Classes, Graphics, Controls, Forms,	111111111111111111111111111111111111111
Dialogs,	
StdCtrls, ExtCtrls;	
,	
type	24 proc
TGDSStdForm = class(TForm)	24 proc

	GDSStdPa
var	GDSStdIm
GridViewForm: TGridViewForm;	GDSLabel:
(3.14 ) 20 (12 0.14)	GDSLabel
implementation	GDSSlogar
Implementation	GDSSlogar
uses RecForm;	ImageBeve
uses rect of m,	private
{\$R *.dfm}	{ Private de
(SK .uim)	public
<> procedure	{ Public de
TGridViewForm.DBGrid1DblClick(	end;
Sender: TObject);	enu,
begin	Non
inherited;	CDSStdEam
Ź	GDSStdFor
RecViewForm.Show;	· 1 / /*
end;	implementati
end.	{\$R *.dfm}
	end.
unit RecForm;	
	\\\\\\
interface	
	unit GridFor
uses	
SysUtils, Windows, Messages,	interface
Classes, Graphics, Controls, Forms,	
Dialogs,	uses
DBCtrls, DB, DBTables, ExtCtrls,	SysUtils, Wi
StdCtrls, Buttons, Mask, Menus,	Classes, Grap
ComCtrls, GdsData;	Dialogs,
ouncurs, ouss um,	Grids, DBG
type	ExtCtrls, Std
TRecViewForm =	GdsData;
class(TStdDataForm)	GusData,
GroupBox2: TPanel;	
OrderNo: TDBEdit;	type TGridViewl
,	I Gria viewi
CustName: TDBEdit;	1 (TC(1D)
C I D / TDDE I'/	
SaleDate: TDBEdit;	DBGrid1:
AmountDue: TDBEdit;	DBGrid1: '
AmountDue: TDBEdit; Label1: TLabel;	DBGrid1: ' <> pi DBGrid1Dbl
AmountDue: TDBEdit; Label1: TLabel; Label2: TLabel;	DBGrid1: ' <> pi DBGrid1Dble TObject);
AmountDue: TDBEdit; Label1: TLabel; Label2: TLabel; Label3: TLabel;	DBGrid1: ' <> pr DBGrid1Dble TObject); private
AmountDue: TDBEdit; Label1: TLabel; Label2: TLabel; Label3: TLabel; Label4: TLabel;	DBGrid1: ' <> pi DBGrid1Dble TObject); private { Private definitions of the content of the c
AmountDue: TDBEdit; Label1: TLabel; Label2: TLabel; Label3: TLabel;	DBGrid1: ' <> pi DBGrid1Dble TObject); private { Private depublic
AmountDue: TDBEdit; Label1: TLabel; Label2: TLabel; Label3: TLabel; Label4: TLabel;	private { Private de

GDSStdPanel: TPanel;
GDSStdImage: TImage;
GDSLabel: TLabel;
GDSLabel2: TLabel;
GDSSloganLabel: TLabel;
Ţ.
GDSSloganLabel2: TLabel;
ImageBevel: TBevel;
private
{ Private declarations }
public
{ Public declarations }
end;
V-14.)
VON
CDCC4dFayrra TCDCC4dFayrra
GDSStdForm: TGDSStdForm;
implementation
{\$R *.dfm}
end.
\\\\\\
1. C. LIE
unit GridForm;
interface
uses
SysUtils, Windows, Messages,
Classes, Graphics, Controls, Forms,
Dialogs,
Grids, DBGrids, DB, DBTables,
ExtCtrls, StdCtrls, Buttons,
GdsData;
type
TGridViewForm =
1 Grid viewi orin
class(TStdDataForm)
class(TStdDataForm)
class(TStdDataForm) DBGrid1: TDBGrid;
class(TStdDataForm) DBGrid1: TDBGrid; <> procedure
class(TStdDataForm)  DBGrid1: TDBGrid;  <> procedure  DBGrid1DblClick(Sender:
class(TStdDataForm)  DBGrid1: TDBGrid;  <> procedure  DBGrid1DblClick(Sender:  TObject);
class(TStdDataForm)  DBGrid1: TDBGrid;  <> procedure  DBGrid1DblClick(Sender:  TObject);  private
class(TStdDataForm)  DBGrid1: TDBGrid;  <> procedure  DBGrid1DblClick(Sender:  TObject);  private  { Private declarations }
class(TStdDataForm)  DBGrid1: TDBGrid;  <> procedure  DBGrid1DblClick(Sender:  TObject);  private
class(TStdDataForm)  DBGrid1: TDBGrid;  <> procedure  DBGrid1DblClick(Sender:  TObject);  private  { Private declarations }
class(TStdDataForm)  DBGrid1: TDBGrid;  <> procedure  DBGrid1DblClick(Sender:  TObject);  private  { Private declarations }  public

Orders: TTable;
Cust: TTable;
OrdersSource: TDataSource;
OrdersOrderNo: TFloatField;
OrdersCustNo: TFloatField;
OrdersSaleDate: TDateTimeField;
OrdersShipDate:
TDateTimeField;
OrdersEmpNo: TIntegerField;
OrdersShipToContact:
TStringField;
OrdersShipToAddr1:
TStringField;
OrdersShipToAddr2:
TStringField;
OrdersShipToCity: TStringField;
OrdersShipToState: TStringField;
OrdersShipToZip: TStringField;
OrdersShipToCountry:
TStringField;
OrdersShipToPhone:
TStringField;
OrdersShipVIA: TStringField;
OrdersPO: TStringField;
OrdersTerms: TStringField;
OrdersPaymentMethod:
TStringField;
OrdersItemsTotal:
TCurrencyField;
OrdersTaxRate: TFloatField;
OrdersFreight: TCurrencyField;
OrdersCustName: TStringField;
OrdersAmountDue:
TCurrencyField;
OrdersAmountPaid:
TCurrencyField;
GroupBox1: TGroupBox;
FilterOnLabel: TLabel;
FilterCriteria: TEdit;
FilterCheckBox: TCheckBox;
NextBtn: TButton;
PriorBtn: TButton;
OrdersTaxAmount:
TCurrencyField;
<> procedure
FilterOnRadioGroupClick(Sender:
TObject);
<> procedure
FormCreate(Sender: TObject);
1 of moreauchemant, 1 object),

DBEdit4: TDBEdit;
Label5: TLabel;
DBEdit1: TDBEdit;
Label9: TLabel;
DBEdit5: TDBEdit;
,
Bevel1: TBevel;
DBNavigator1: TDBNavigator;
Panel1: TPanel;
OnCredit: TDBCheckBox;
Label11: TLabel;
DBEdit7: TDBEdit;
DBEdit8: TDBEdit;
Label12: TLabel;
private
{ Private declarations }
·
public (Park's deslarations)
{ Public declarations }
end;
var
RecViewForm: TRecViewForm;
implementation
•
{\$R *.dfm}
(4
end.
Citu.
\\\\\\\
u CLD /
unit GdsData;
interface
uses
SysUtils, Windows, Messages,
Classes, Graphics, Controls, Forms,
Dialogs,
Buttons, StdCtrls, ExtCtrls, Grids,
DBGrids, DB, DBTables, Mask,
DBCtrls, GdsStd;
type
TStdDataForm =
TStdDataForm =
TStdDataForm = class(TGDSStdForm) StdCtrlPanel: TPanel;
TStdDataForm = class(TGDSStdForm)

O I O EU D I	
Orders.OnFilterRecord :=	<pre>&lt;&gt; procedure OndowsEithonOn Data Code </pre>
OrdersFilterOnDate; FilterCriteria.Text :=	OrdersFilterOnDate(DataSet:
	TDataSet; var Accept: Boolean); <> procedure
DateToStr(FLastDate);	OrdersFilterOnAmount(DataSet:
end;	· · · · · · · · · · · · · · · · · · ·
1: begin	TDataSet; var Accept: Boolean);
Orders.OnFilterRecord :=	<> procedure
OrdersFilterOnAmount;	OrdersCalcFields(DataSet: TDataSet);
FilterCriteria.Text :=	
FloatToStr(FLastAmount);	<pre>&lt;&gt; procedure FilterCheckBoxClick(Sender:</pre>
end;	TObject);
end;	<b>V</b> //
ActiveControl := FilterCriteria;	<pre></pre>
end;	PriorBtnClick(Sender: TObject);
if Orders.Filtered then	<pre>&lt;&gt; procedure NovtPtnClick(Sondow TObicot);</pre>
Orders.Refresh;	NextBtnClick(Sender: TObject);
end;	<> procedure
	FilterCriteriaExit(Sender: TObject);
<> procedure	<> procedure
TStdDataForm.FormCreate(Sender:	FilterCriteriaKeyPress(Sender:
TObject);	TObject; var Key: Char);
begin	protected
inherited;	FLastAmount: Double;
FLastDate := EncodeDate(1995, 1,	FLastDate: TDateTime;
1);	function CalcAmountDue:
FLastAmount := 1000;	Double;
FilterOnRadioGroup.ItemIndex :=	<> procedure
0;	ConvertFilterCriteria;
end;	end;
{ Calculate the value of AmountDue.	var
Used in the OnCalcFields	StdDataForm: TStdDataForm;
and OnFilterRecord event	
handlers. }	implementation
function	<b>{\$R *.dfm}</b>
TStdDataForm.CalcAmountDue:	
Double;	<> procedure
begin	TStdDataForm.FilterOnRadioGrou
Result :=	pClick(Sender: TObject);
OrdersItemsTotal.Value * (1.0 +	begin
OrdersTaxRate.Value / 100) +	inherited;
OrdersFreight.Value -	with FilterOnRadioGroup do
OrdersAmountPaid.Value;	begin
end;	FilterOnLabel.Caption :=
- ")	Format('Records where %S >=',
{ Convert the FilterCriteria text into	[Items[ItemIndex]]);
a Date or Float. This value	case ItemIndex of
will be used in the OnFilterRecord	0: begin
min be used in the Onl'littli Netul U	~ · ~ · · · · · · · · · · · · · · · · ·

if its amount is greater than or equal to the specified filter criteria. Note that calculated and lookup fields are undefined in an OnFilterRecord event handler. }	fly. Include in the dataset	callback instead of using the
string does not need to be   converted each time the event is tringered. }		
criteria. Note that calculated and lookup fields are undefined in an OnFilterRecord event handler. }	_	•
lookup fields are undefined   in an OnFilterRecord event   handler. }	-	
in an OnFilterRecord event handler.}		
Annalder.   ConvertFilterCriteria   ConvertFilterCri	-	triggered.
TStdDataForm.OrdersFilterOnAmount(DataSet: TDataSet; var Accept: Boolean); begin inherited; Accept: = CalcAmountDue >= FLastAmount; end; {Include this order in the dataset if its date is greater than or equal to the specified filter criteria.}  <> procedure TStdDataForm.OrdersFilterOnDate (DataSet: TDataSet; var Accept: Boolean); begin inherited; Accept:= OrdersSaleDate.Value >= FLastDate; end; > procedure TStdDataForm.OrdersCalcFields(DataSet: TDataSet); begin inherited; OrdersTaxAmount.Value := OrdersItemsTotal.Value *= OrdersItemsTotal.Value *= OrdersTaxAate.Value / 100); OrdersAmountDue.Value := CalcAmountDue; end; { Store contents of filter criteria from edit control}		< > procedure
a: begin inherited; Accept := CalcAmountDue >= FLastAmount; end; {Include this order in the dataset if its date is greater than or equal to the specified filter criteria.}  **StrToFloat(FilterCriteria.Text); if JataSet: TDataSet; var Accept: Boolean); begin inherited; Accept := OrdersSaleDate.Value >= FLastDate; end;  **C> procedure TStdDataForm.OrdersCalcFields(DataSet: TDataSet); begin inherited; OrdersTaxAmount.Value := OrdersItemsTotal.Value ** OrdersTaxAmountDue.Value := CalcAmountDue, Value := CalcAmountDue; end;  { Store contents of filter criteria from edit control}  **Store contents of filter criteria from edit control}  **ConvertFilterCriteria.Text ** "then case FilterOnRadioGroup.ItemIndex of 0: FLastDate := StrToDate(FilterCriteria.Text);  !: FLastAmount := StrToDate(FilterCriteria.Text);  !: FLastAmount := StrToDate(FilterCriteria.Text);  ! FLastAmount := StrToDate(FilterCriteria.Text);  ! FLastAmount := StrToBationCriteria.Text *" then  0: FLastDate := StrToDate(FilterCriteria.Text);  !: FLastAmount := StrToBationCriteria.Text *" then  0: FLastDate := StrToDate(FilterCriteria.Text);  ! FLastAmount := StrToBationCriteria.Text);  ! FLastAmount := StrToBationCriteria.Text *" then  0: FLastDate := StrToBationCriteria.Text **    StrToBationCriteria.Text **   StrToBloat(FilterCriteria.Text);    Conders.FilterCheck **   Tyt to convert the filter criteria whenever the edit control looses focus, or the user presses enter }    StrObationCriteria.Text **   StrObationCriteria.Text **   StrToBloat(FilterCriteria.Text **   StrToBloat(FilterCriteria.Text **   StrToBloa	,	
unt(DataSet: TDataSet; var Accept: Boolean); begin inherited; Accept := CalcAmountDuc >= FLastAmount; end;  {Include this order in the dataset if its date is greater than or equal to the specified filter criteria.}  **Comparity of the specified filter criteria.}  **Var Accept: Boolean);  begin inherited;  Accept := OrdersSaleDate.Value >= FLastDate; end;  **Comparity of the specified specified specified inherited;  CordersTaxAmount.Value := OrdersTaxAmount.Value := OrdersTaxAmountDue.Value := CalcAmountDue.Value := CalcAmo	I - I	
var Accept: Boolean);   begin   inherited;   Accept := CalcAmountDue >=   StrToBtate :=   St		
case   FilterOnRadioGroup.ItemIndex of		
inherited; Accept := CalcAmountDue >= FLastAmount; end;  { Include this order in the dataset if its date is greater than or equal to the specified filter criteria. } > procedure TStdDataForm.OrdersFilterOnDate (DataSet: TDataSet; var Accept := OrdersSaleDate.Value >= FLastDate; end; > procedure TStdDataForm.OrdersCalcFields(DataSet: TDataSet); begin inherited; > procedure TStdDataForm.OrdersCalcFields(DataSet: TDataSet); begin inherited;  Orders TaxAmount.Value := OrdersItemsTotal.Value * (OrdersTaxRate.Value / 100); OrdersAmountDue.Value := CalcAmountDue; end;  {Store contents of filter criteria from edit control }  Store contents of filter criteria from edit control }  Store contents of filter criteria from edit control }  Store contents of filter criteria from edit control }  Store contents of filter criteria from edit control }  Store contents of filter criteria from edit control }  ConvertFilterCriteria.Text); end;  if Orders.FilterCd then Orders.Refresh; end;  {Try to convert the filter criteria whenever the edit control looses focus, or the user presses enter }  StrdDataForm.FilterCriteriaExit(Se nder: TObject);  begin inherited;  ConvertFilterCriteria.Text); end;  if Orders.Filtered then Orders.Refresh; end;  Convert the filter criteria whenever the edit control looses focus, or the user presses enter }  StrdDataForm.FilterCriteriaExit(Se nder: TObject);  var Key: Char); begin inherited;  ConvertFilterCriteria; if Key = #13 then begin  ConvertFilterCriteria; Key := #0 end; end;		
Accept := CalcAmountDue >= FLastAmount; end;  {Include this order in the dataset if its date is greater than or equal to the specified filter criteria. }  **Common Procedure** TStdDataForm.OrdersFilterOnDate** (DataSet: TDataSet; var Accept: Boolean); begin inherited; Accept := OrdersSaleDate.Value >= FLastDate; end;  **Common Procedure** TStdDataForm.OrdersCalcFields(DataSet: TDataSet); begin inherited; OrdersTaxAmount.Value := OrdersItemsTotal.Value * (OrdersTaxAmountDue.Value := CalcAmountDue; end;  { Store contents of filter criteria from edit control }  **StToFloat(FilterCriteria.Text);  1: FLastAmount := StrToDate(FilterCriteria.Text); end;  if Orders.Filtered then Orders.Refresh; end;  { Try to convert the filter criteria whenever the edit control looses focus, or the user presses enter }  **Common FilterCriteriaExit(Se nder: TObject);  begin inherited; ConvertFilterCriteria; end;  **ConvertFilterCriteria from ExidDataForm.FilterCriteriaKeyPress(Sender: TObject; var Key: Char);  begin inherited;  **ConvertFilterCriteria from ExidDataForm.FilterCriteria from ExidData		
FLastAmount; end;  Include this order in the dataset if its date is greater  than or equal to the specified filter criteria. }	,	-
Include this order in the dataset if its date is greater than or equal to the specified filter criteria. } > procedure TStdDataForm.OrdersFilterOnDate (DataSet: TDataSet; var Accept: Boolean); begin inherited; Accept := OrdersSaleDate.Value >= FLastDate; end; > procedure TStdDataForm.OrdersCalcFields(D ataSet: TDataSet); begin inherited; OrdersTaxAmount.Value := OrdersTaxAmount.Value := OrdersTaxAmount.Value := CalcAmountDue.Value := CalcAmo		
StrToFloat(FilterCriteria.Text);  { Include this order in the dataset if its date is greater  than or equal to the specified filter criteria.}  <> procedure  TStdDataForm.OrdersFilterOnDate (DataSet: TDataSet; var Accept: Boolean); begin inherited;  Accept := OrdersSaleDate.Value >= FLastDate; end; > procedure  TStdDataForm.OrdersCalcFields(DataSet: TDataSet); begin inherited;  OrdersTaxAmount.Value := OrdersItemsTotal.Value * (OrdersTaxRate.Value / 100); OrdersTaxRate.Value / 100); OrdersAmountDue.Value := CalcAmountDue; end;  { Store contents of filter criteria from edit control } > procedure  TStdDataForm.FilterCriteria(; if Key = #13 then begin  ConvertFilterCriteria; Key := #0 end; end; end; end;	· · · · · · · · · · · · · · · · · · ·	
Include this order in the dataset if its date is greater	enu;	
its date is greater than or equal to the specified filter criteria. }	(Include this and an in the little of the	
than or equal to the specified filter criteria. }  <> procedure TStdDataForm.OrdersFilterOnDate (DataSet: TDataSet; var Accept: Boolean); begin inherited; Accept := OrdersSaleDate.Value >= FLastDate; end; > procedure TStdDataForm.OrdersCalcFields(DataSet: TDataSet); begin inherited; OrdersTaxAmount.Value := OrdersTaxAmount.Value := OrdersTaxRate.Value / 100); OrdersAmountDue.Value := CalcAmountDue, cend;  { Store contents of filter criteria from edit control } > procedure TStdDataForm.FilterCriteriaKeyPr ess(Sender: TObject; var Key: Char); begin inherited; ConvertFilterCriteriaKeyPr ess(Sender: TObject; var Key: Char); begin inherited; ConvertFilterCriteriaKeyPr ess(Sender: TObject; var Key: Char); begin inherited; convertFilterCriteriaKeyPr ess(Sender: TObject; var Key: Char); begin inherited; if Key = #13 then begin ConvertFilterCriteria; Key := #0 end;		,
criteria. }  <> procedure TStdDataForm.OrdersFilterOnDate (DataSet: TDataSet; var Accept: Boolean); begin inherited; Accept := OrdersSaleDate.Value >= FLastDate; end; > procedure TStdDataForm.OrdersCalcFields(D ataSet: TDataSet); begin inherited; OrdersTaxAmount.Value := OrdersTaxAmount.Value := ColcAmountDue.Value := CalcAmountDue; end;  { Store contents of filter criteria from edit control } > procedure TStdDataForm.FilterCriteriaKeyPr ess(Sender: TObject; var Key: Char); begin inherited; ConvertFilterCriteriaKeyPr ess(Sender: TObject; var Key: Char); begin ConvertFilterCriteriaKeyPr ess(Sender: TObject; var Key: Char); begin ConvertFilterCriteria Key := #13 then begin ConvertFilterCriteria; key := #0 end; end;		
> procedure TStdDataForm.OrdersFilterOnDate (DataSet: TDataSet;  var Accept: Boolean);  begin inherited; Accept := OrdersSaleDate.Value >= FLastDate; end; > procedure TStdDataForm.OrdersCalcFields(D ataSet: TDataSet); begin inherited; OrdersTaxAmount.Value := OrdersTaxAmount.Value * (OrdersTaxAte.Value / 100); OrdersAmountDue.Value := CalcAmountDue; end;  { Store contents of filter criteria from edit control } > procedure TStdDataForm.FilterCriteriaKeyPr ess(Sender: TObject; var Key: Char); begin inherited; ConvertFilterCriteriaKeyPr ess(Sender: TObject; var Key: Char); begin ConvertFilterCriteria inherited; if Key = #13 then begin ConvertFilterCriteria; Key := #0 end; end;		
TStdDataForm.OrdersFilterOnDate (DataSet: TDataSet;  var Accept: Boolean); begin inherited;  Accept:= OrdersSaleDate.Value >= FLastDate; end; > procedure TStdDataForm.OrdersCalcFields(DataSet: TDataSet); begin inherited;  ConvertFilterCriteria; end;  OrdersTaxAmount.Value := OrdersTaxAmountDue.Value := CalcAmountDue; end;  {Store contents of filter criteria from edit control}  Store contents of filter criteria from edit control} > procedure TStdDataForm.FilterCriteria; key:=#0 end;	,	ena;
(DataSet: TDataSet;   var Accept: Boolean);   begin   looses focus, or the user presses enter }   looses focus, or the user presses enter }   coses focus enter   coses focus enter   coses	_	
looses focus, or the user presses enter }   begin		7 9
begin		
inherited; Accept := OrdersSaleDate.Value >= FLastDate; end; > procedure TStdDataForm.OrdersCalcFields(D ataSet: TDataSet); begin inherited; OrdersTaxAmount.Value := OrdersItemsTotal.Value * (OrdersTaxRate.Value / 100); OrdersAmountDue.Value := CalcAmountDue; end;  { Store contents of filter criteria from edit control }  { Store contents of filter criteria from edit control }  { Store contents of filter Criteria from edit control }  { Store contents of filter Criteria from edit control }  CalcAmountDue; end;  Accept := OrdersSaleDate.Value >= TStdDataForm.FilterCriteriaExit(Se nder: TObject);  begin  ConvertFilterCriteriaKeyPr ess(Sender: TObject; var Key: Char); begin inherited; if Key = #13 then begin ConvertFilterCriteria; Key := #0 end; end;		•
Accept := OrdersSaleDate.Value >= FLastDate; end; > procedure TStdDataForm.OrdersCalcFields(D ataSet: TDataSet); begin inherited; OrdersTaxAmount.Value := OrdersItemsTotal.Value * (OrdersTaxRate.Value / 100); OrdersAmountDue.Value := CalcAmountDue; end;  { Store contents of filter criteria from edit control }  { Store contents of filter Criteria from edit control }  { Store TObject);		enter }
FLastDate; end; > procedure TStdDataForm.OrdersCalcFields(D ataSet: TDataSet); begin inherited; OrdersTaxAmount.Value := OrdersItemsTotal.Value * (OrdersTaxRate.Value / 100); OrdersAmountDue.Value := CalcAmountDue; end;  { Store contents of filter criteria from edit control }  { Store contents of filter Criteria from edit control } > procedure TStdDataForm.FilterCriteriaExit(Se nder: TObject);  begin > procedure TStdDataForm.FilterCriteriaKeyPr ess(Sender: TObject;  var Key: Char); begin inherited; if Key = #13 then begin  ConvertFilterCriteria; Key := #0 end; end; end;	,	
end; > procedure TStdDataForm.OrdersCalcFields(D ataSet: TDataSet); begin inherited; OrdersTaxAmount.Value := OrdersItemsTotal.Value * (OrdersTaxRate.Value / 100); OrdersAmountDue.Value := CalcAmountDue; end;  { Store contents of filter criteria from edit control } > procedure TStdDataForm.FilterCriteriaKeyPr ess(Sender: TObject; var Key: Char); begin inherited; var Key: Char); begin inherited; if Key = #13 then begin  ConvertFilterCriteria; Key := #0 end; end;	I = I	=
begin   inherited;   ConvertFilterCriteria;   end;	,	
<pre>&gt; procedure TStdDataForm.OrdersCalcFields(D ataSet: TDataSet); begin inherited; OrdersTaxAmount.Value := OrdersItemsTotal.Value * (OrdersTaxRate.Value / 100); OrdersAmountDue.Value := CalcAmountDue; end; {Store contents of filter criteria from edit control } &lt;&gt; procedure TStdDataForm.FilterCriteria; from edit control } &lt;&gt; procedure TStdDataForm.FilterCheckBoxClic k(Sender: TObject);</pre> inherited; ConvertFilterCriteria; var Key: Char); begin inherited; ConvertFilterCriteria(FilterCriteria) convertFi	end;	* :
TStdDataForm.OrdersCalcFields(D ataSet: TDataSet); begin inherited; OrdersTaxAmount.Value := OrdersItemsTotal.Value * (OrdersTaxRate.Value / 100); OrdersAmountDue.Value := CalcAmountDue; end;  { Store contents of filter criteria from edit control }  <> procedure TStdDataForm.FilterCriteria; end; > begin inherited; if Key = #13 then begin ConvertFilterCriteria; fey = #13 then begin ConvertFilterCriteria; if Key = #13 then begin ConvertFilterCriteria; end;		
ataSet: TDataSet); begin inherited; OrdersTaxAmount.Value := OrdersItemsTotal.Value * (OrdersTaxRate.Value / 100); OrdersAmountDue.Value := CalcAmountDue; end;  Store contents of filter criteria from edit control }  <> procedure TStdDataForm.FilterCriteria(; if Key = #13 then begin ConvertFilterCriteria; Key := #0 end; end; end;	=	
begin inherited; OrdersTaxAmount.Value := OrdersItemsTotal.Value * (OrdersTaxRate.Value / 100); OrdersAmountDue.Value := CalcAmountDue; end;  { Store contents of filter criteria from edit control }  <> procedure TStdDataForm.FilterCriteriaKeyPr ess(Sender: TObject; var Key: Char); begin inherited; if Key = #13 then begin ConvertFilterCriteria; Key := #0 end; end; end;	`	
inherited; OrdersTaxAmount.Value := OrdersItemsTotal.Value * (OrdersTaxRate.Value / 100); OrdersAmountDue.Value := CalcAmountDue; end;  { Store contents of filter criteria from edit control }  <> procedure TStdDataForm.FilterCriteriaKeyPr ess(Sender: TObject; var Key: Char); begin inherited; if Key = #13 then begin ConvertFilterCriteria; Key := #0 end; end; end;		end;
OrdersTaxAmount.Value := OrdersItemsTotal.Value * (OrdersTaxRate.Value / 100); OrdersAmountDue.Value := CalcAmountDue; end;  { Store contents of filter criteria from edit control }  <> procedure TStdDataForm.FilterCriteriaKeyPr ess(Sender: TObject; var Key: Char); begin inherited; if Key = #13 then begin ConvertFilterCriteria; Key := #0 end; end; end;	C	
OrdersItemsTotal.Value * (OrdersTaxRate.Value / 100); OrdersAmountDue.Value := CalcAmountDue; end;  { Store contents of filter criteria from edit control }  <> procedure TStdDataForm.FilterCheckBoxClic k(Sender: TObject);    Store contents of filter criteria from edit control }  > procedure    > procedure    > procedure    > procedure    > procedure	,	<u> </u>
(OrdersTaxRate.Value / 100); OrdersAmountDue.Value := CalcAmountDue; end;  { Store contents of filter criteria from edit control }  <> procedure TStdDataForm.FilterCheckBoxClic k(Sender: TObject);  var Key: Char); begin inherited; if Key = #13 then begin ConvertFilterCriteria; Key := #0 end; end;		
OrdersAmountDue.Value := CalcAmountDue; end;  [Store contents of filter criteria from edit control } > procedure TStdDataForm.FilterCheckBoxClic k(Sender: TObject);    begin inherited;   if Key = #13 then     begin     ConvertFilterCriteria;   Key := #0     end;   end;		• •
CalcAmountDue; end;  inherited; if Key = #13 then begin  ConvertFilterCriteria; from edit control }  ConvertFilterCriteria; Key := #0 end; end;  TStdDataForm.FilterCheckBoxClic k(Sender: TObject);		,
end;  { Store contents of filter criteria from edit control }  <> procedure  TStdDataForm.FilterCheckBoxClic k(Sender: TObject);  if Key = #13 then begin  ConvertFilterCriteria; Key := #0 end; end;		
Store contents of filter criteria   ConvertFilterCriteria;   Key := #0   end;   end;   k(Sender: TObject);	· · · · · · · · · · · · · · · · · · ·	,
{ Store contents of filter criteria from edit control }  <> procedure  TStdDataForm.FilterCheckBoxClic k(Sender: TObject);  ConvertFilterCriteria;  Key := #0  end;  end;	end;	if Key = #13 then
from edit control }  <> procedure  TStdDataForm.FilterCheckBoxClic k(Sender: TObject);  Key := #0 end; end; end;		
<pre>&lt;&gt; procedure TStdDataForm.FilterCheckBoxClic k(Sender: TObject);</pre> end; end;	,	ConvertFilterCriteria;
TStdDataForm.FilterCheckBoxClic end;	,	Key := #0
k(Sender: TObject);	<u>-</u>	end;
		end;
Segin	k(Sender: TObject);	
	begin	{ Calculate this order's total on the

CUL ATT 1 1 AGG
ClientHeight = 322
ClientWidth = 480
PixelsPerInch = 96
TextHeight = 13
inherited GDSStdPanel: TPanel
Width = 480
end
inherited StdCtrlPanel: TPanel
Width = 480
inherited PriorBtn: TButton
Left = 385
end
end
<> Object GroupBox2:
TPanel [2]
Left = 0
Top = 161
Width = 480
Height = 161
Align = alClient
BevelOuter = bvLowered
TabOrder = 2
<> Object Label1: TLabel
Left = 8
Top = 42
Width = 46
Height = 13
Caption = 'Order No:'
end
<> Object Label2: TLabel
Left = 8
Top = 68
<b>Width</b> = 47
Height = 13
Caption = 'Customer:'
end
<> Object Label3: TLabel
Left = 131
Top = 41
Width = 50
Height = 13
Caption = 'Sale Date:'
end
<> Object Label4: TLabel  Left = 300
Top = 126
Width = 62
Height = 13

inherited;
ConvertFilterCriteria;
Orders.Filtered :=
FilterCheckBox.Checked;
NextBtn.Enabled := not
FilterCheckBox.Checked;
PriorBtn.Enabled := not
FilterCheckBox.Checked;
end;
{ Button handlers for new filter-
oriented dataset navigation methods
}
<> procedure
TStdDataForm.PriorBtnClick(Sende
r: TObject);
begin
inherited;
ConvertFilterCriteria;
Orders.FindPrior;
end;
<> procedure
TStdDataForm.NextBtnClick(Sende
r: TObject);
begin
inherited;
ConvertFilterCriteria;
Orders.FindNext;
end;
end.
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

<> Objects27	Ī
الملف النصي لبرنامج	
(GdsDemo)	
عامدا	5
علاء الدين اللباد	
۶۹۳۳ ۱۸۵۸۵۶۶۰۰ ۱۸۹۳۳ ۱۸۹۵۸۵۶۶۰۰	
inherited RecViewForm:	
TRecViewForm	
Caption = 'Single Record View'	

DataSource = OrdersSource
TabOrder = 0
end
<> Object CustName:
TDBEdit
Left = 66
Top = 65
Width = 187
Height = 21
DataField = 'CustName'
DataSource = OrdersSource
TabOrder = 1
end
<> Object SaleDate:
TDBEdit
Left = 186
Top = 39
Width = 67
Height = 21
DataField = 'SaleDate'
DataSource = OrdersSource
TabOrder = 2
end
<> Object AmountDue:
TDBEdit
Left = 376
Top = 123
Width = 81
Height = 21
DataField = 'Amount'
DataSource = OrdersSource
TabOrder = 3
end
<> Object DBEdit3:
TDBEdit
Left = 376
Top = 15
Width = 81
Height = 21
DataField = 'ItemsTotal'
DataSource = OrdersSource
TabOrder = 4
end
<> Object DBEdit4:
TDBEdit
Left = 376
Top = 87
Width = 81

Caption = 'Amount Due:'
end
<> Object Label7: TLabel
Left = 300
Top = 19
Width = 55
Height = 13
Caption = 'Items Total:'
FocusControl = DBEdit3
end
<> Object Label8: TLabel
Left = 300
Top = 92
Width = 63
Height = 13
Caption = 'Amount Paid:'
FocusControl = DBEdit4
end
<> Object Label5: TLabel
Left = 300
Top = 67
<b>Width = 35</b>
Height = 13
Caption = 'Freight:'
FocusControl = DBEdit1
end
<> Object Label9: TLabel
Left = 300
Top = 43
Width = 21
Height = 13
Caption = 'Tax:'
FocusControl = DBEdit5
end
<> Object Bevel1: TBevel
Left = 309
Top = 115
Width = 148
Height = 2
Shape = bsTopLine
end
<> Object OrderNo:
TDBEdit
Left = 66
Top = 39
Width = 50
Height = 21
DataField = 'OrderNo'
Datarieiu – Oruerivo

Height = 13
Caption = 'Terms:'
FocusControl = DBEdit7
end
<> Object Label12: TLabel
Left = 23
Top = 10
Width = 18
Height = 13
Caption = 'PO:' FocusControl = DBEdit8
end
<> Object OnCredit:
TDBCheckBox
Left = 96
Top = 32
Width = 77
Height = 17
Caption = 'On Credit'
DataField = 'PaymentMethod'
DataSource = OrdersSource
TabOrder = 0
ValueChecked = 'Credit'
ValueUnchecked =
'Visa;Cash;Check;MC;Amex'
end
<> Object DBEdit7:
TDBEdit
Left = 44
Top = 30
Width = 40
Height = 21
DataField = 'Terms'
DataSource = OrdersSource
TabOrder = 1
end
<> Object DBEdit8:
TDBEdit
Left = 44
Top = 6
Width = 113
Height = 21
DataField = 'PO'
DataSource = OrdersSource
TabOrder = 2
end
end
end

Height = 21
DataField = 'AmountPaid'
DataSource = OrdersSource
TabOrder = 5
end
<> Object DBEdit1:
TDBEdit
Left = 376
Top = 63
Width = 81
Height = 21
DataField = 'Freight'
DataSource = OrdersSource
TabOrder = 6
end
<> Object DBEdit5:
TDBEdit
Left = 376
Top = 39
Width = 81
Height = 21
DataField = 'TaxAmount'
DataSource = OrdersSource
TabOrder = 7
end
<pre>&lt;&gt; Object DBNavigator1: TDPNavigator</pre>
TDBNavigator
Left = 107
Top = 10
Width = 88
Height = 21
DataSource = OrdersSource
VisibleButtons = [nbFirst,
nbPrior, nbNext, nbLast]
TabOrder = 8
end
<> Object Panel1: TPanel
Left = 71
Top = 92
<b>Width</b> = 177
Height = 57
BevelInner = bvRaised
BevelOuter = bvLowered
TabOrder = 9
<> Object Label11: TLabel
Left = 7
Top = 34
Width = 32
· · · · · · · · · · · · · · · · · · ·

Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
OnCloseQuery = FormCloseQuery
OnCreate = FormCreate
OnDestroy = FormDestroy
PixelsPerInch = 96
TextHeight = 13
<> object Label1: TLabel
Left = 16
Top = 16
Width = 74
Height = 13
Caption = '&Database Alias:'
FocusControl = AliasCombo
end
<> object Label2: TLabel
Left = 16
Top = 56
Width = 31
Height = 13
Caption = '&Query:'
FocusControl = QueryEdit
end
<> object Label3: TLabel
Left = 152
Top = 16
Width = 56
Height = 13
Caption = '&User Name:'
FocusControl = NameEdit
end
<> object Label4: TLabel
Left = 304
Top = 16
Width = 49
Height = 13
Caption = '&Password:'
FocusControl = PasswordEdit
end
<> object Label5: TLabel
Left = 15
Top = 259
Width = 72
Height = 13
Caption = '&Retrieve query:'
FocusControl =

End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
inherited GridViewForm:
TGridViewForm
Caption = 'Grid View'
PixelsPerInch = 96
TextHeight = 13
<> Object DBGrid1:
TDBGrid [2]
Left = 0
Top = 161
Width = 460
Height = 159
Align = alClient
DataSource = OrdersSource
TabOrder = 2
TitleFont.Charset =
DEFAULT_CHARSET
TitleFont.Color = clBlack
TitleFont.Height = -11
TitleFont.Name = 'MS Sans Serif'
TitleFont.Style = []
OnDblClick = DBGrid1DblClick
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

<> object \$31
الملف النصي لبرنامج
(BkQuery)
عداد
علاء الدين اللباد
واتف ۷۹۲۵۷۵۲۹۰
<> object AdhocForm:
TAdhocForm
Left = 144
Top = 297
BorderStyle = bsSingle
Caption = 'Adhoc Background
Query Demo'
ClientHeight = 297
ClientWidth = 545

TabOrder = 2
Text = 'masterkey'
end
<> object ExecuteBtn:
TButton
Left = 464
Top = 8
Width = 75
Height = 25
Caption = '&Execute'
Default = True
TabOrder = 5
OnClick = ExecuteBtnClick
end
<> object CloseBtn: TButton
Left = 464
Top = 134
Width = 75
Height = 25
Cancel = True
Caption = '&Close'
TabOrder = 6
OnClick = CloseBtnClick
end
<> object SavedQueryCombo:
TComboBox
Left = 96
Top = 256
Width = 345
Height = 21
Style = csDropDownList
ItemHeight = 13
TabOrder = 4
OnChange =
SavedQueryComboChange
end
<> object SaveBtn: TButton
Left = 464
Top = 72
Width = 75
Height = 25
Caption = '&Save'
TabOrder = 7
OnClick = SaveBtnClick
end
<> object SaveAsBtn:
TButton
Left = 464
·

SavedQueryCombo
end
<> object Bevel1: TBevel
Left = 8
Top = 8
Width = 449
Height = 281
Shape = bsFrame
end
<> object AliasCombo:
TComboBox
Left = 16
Top = 32 Width = 129
Height = 21
ItemHeight = 13
TabOrder = 0
Text = 'AliasCombo'
end
<> object QueryEdit: TMemo
Left = 16
Top = 72
Width = 425
Height = 177
Lines.Strings = (
'select Company,
Sum(ItemsTotal) -
Sum(AmountPaid) as AmountDue '
' from customer, orders '
' where Customer.CustNo =
Orders.CustNo'
' group by Company')
TabOrder = 3
end
<> object NameEdit: TEdit
Left = 152
Top = 32
Width = 145
Height = 21
TabOrder = 1
end
<> object PasswordEdit:
TEdit
Left = 304
Top = 32
Width = 137
Height = 21
PasswordChar = '*'

Align = alBottom
Caption = 'Opening'
WordWrap = True
end
<> object DBGrid1: TDBGrid
Left = 0
Top = 13
Width = 435
Height = 272
Align = alClient
DataSource = DataSource
TabOrder = 0
TitleFont.Charset =
DEFAULT CHARSET
TitleFont.Color = clWindowText
TitleFont.Height = -11
TitleFont.Name = 'MS Sans Serif'
TitleFont.Style = []
end
<> object DataSource:
TDataSource
Left = 112
Top = 248
end
<> object Query: TQuery
Left = 80
Top = 248
end
<> object Session: TSession
Left = 16
Top = 248
end
<> object Database:
TDatabase
LoginPrompt = False
SessionName = 'Default'
Left = 48
Top = 248
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
3
<> object SaveQueryAs:
TSaveQueryAs
Left = 111
Top = 149
-
BorderStyle = bsDialog

Top = 104
Width = 75
Height = 25
Caption = 'Save &as'
TabOrder = 8
OnClick = SaveAsBtnClick
end
<> object NewBtn: TButton
Left = 464
Top = 40
Width = 75
Height = 25
Caption = '&New'
TabOrder = 9
OnClick = NewBtnClick
end
End
\\\\\\\\\\\
2
<> object QueryForm:
TQueryForm
Left = 170
Top = 114
Width = 443
Height = 332
Caption = 'Background Query'
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poDefault
PixelsPerInch = 96
TextHeight = 13
<> object QueryLabel:
TLabel Left = 0
Top = 0
Width = 435
Height = 13
Align = alTop
Caption = '[Query text]'
WordWrap = True
end
<> object StatusLine: TLabel
Left = 0
Top = 285
Width = 435
Height = 13

ModalResult = 2	
TabOrder = 2	
end	
End	
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	

الهلف التنفيذي لبرناهم
Procedure37
BkQuery)
عامدا
علاء الدين اللباد
هاتف ۷۳۷۱۹۹۱
{ Demostrates how to execute a
query in a background thread. This
files contains the main user
interface for this program. The background
query code is in ResltFrm }
query couc is in Resid in j
unit QueryFrm;
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, IniFiles, ExtCtrls,
DBTables;
type
TAdhocForm = class(TForm)
AliasCombo: TComboBox;
Label1: TLabel;
QueryEdit: TMemo;
Label2: TLabel;
NameEdit: TEdit;
·
PasswordEdit: TEdit;
PasswordEdit: TEdit; Label3: TLabel;
PasswordEdit: TEdit;

Caption = 'Save query as'
ClientHeight = 102
ClientWidth = 343
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
PixelsPerInch = 96
TextHeight = 13
<> object Bevel1: TBevel
Left = 8
Top = 8
Width = 329
Height = 57
Shape = bsFrame
end
<> object Label1: TLabel
Left = 16
Top = 11
Width = 60
Height = 13
Caption = 'Query name:'
end
<> object NameEdit: TEdit
Left = 16
Top = 27
Width = 305
Height = 21
TabOrder = 0
OnChange = NameEditChange
end
<> object OKBtn: TButton
Left = 184
Top = 72
Width = 75
Height = 25
Caption = 'OK'
Default = True
ModalResult = 1
TabOrder = 1
end
<> object CancelBtn: TButton Left = 264
Top = 72
Width = 75
Height = 25
Cancel = True
Caption = 'Cancel'

function StrToIniStr(const Str:
string): string;
var
Buffer: array[04095] of Char;
B, S: PChar;
begin
if Length(Str) > SizeOf(Buffer)
then
raise Exception.Create('String to
large to save in INI file');
S := PChar(Str);
B := Buffer;
while S^ <> #0 do
case S^ of
#13, #10:
begin
if (S^ = #13) and (S[1] = #10)
then Inc(S)
else if (S^ = #10) and (S[1] =
#13) then Inc(S);
B^ := '\';
Inc(B);
B^ := 'n';
Inc(B);
Inc(S);
end;
else
$\mathbf{B}^{\wedge} := \mathbf{S}^{\wedge};$
Inc(B);
Inc(S);
end;
B^ := #0;
Result := Buffer;
end;
function IniStrToStr(const Str:
string): string;
var
Buffer: array[04095] of Char;
B, S: PChar;
begin
if Length(Str) > SizeOf(Buffer)
then
raise Exception.Create('String to
read from an INI file');
S := PChar(Str);
B := Buffer;

SavedQueryCombo: TComboBox;
Label5: TLabel;
SaveBtn: TButton;
SaveAsBtn: TButton;
NewBtn: TButton;
Bevel1: TBevel;
<> procedure
FormCreate(Sender: TObject);
<> procedure
CloseBtnClick(Sender: TObject);
<> procedure
ExecuteBtnClick(Sender: TObject);
<> procedure
FormDestroy(Sender: TObject);
<> procedure
NewBtnClick(Sender: TObject);
<> procedure
SaveBtnClick(Sender: TObject);
<> procedure
SaveAsBtnClick(Sender: TObject);
<> procedure
SavedQueryComboChange(Sender:
TObject);
<> procedure
FormCloseQuery(Sender: TObject;
var CanClose: Boolean);
private
{ Private declarations }
QueryName, OldAlias: string;
SavedQueries: TIniFile;
Unnamed: Boolean;
function IsModified: Boolean;
function CheckModified: Boolean;
<> procedure Unmodify;
<> procedure Officially;
<pre>&lt;&gt; procedure SaveQuery;</pre>
<> procedure SaveQueryAs;
public (Public declarations)
{ Public declarations }
end;
Aller France TAiler France
AdhocForm: TAdhocForm;
implementation
DD D 1/E C C A
uses DB, ResltFrm, SaveQAs;
(OD + 16 )
{\$R *.dfm}

QueryEdit.Modified := False;	while S
end;	if (S[0
	begin
<> procedure	B^:
TAdhocForm.ReadQuery;	Inc(
begin	<b>B</b> ^:
if not CheckModified then Exit;	Inc(
with SavedQueries do	Inc(
begin	Inc(
QueryName :=	end
SavedQueryCombo.Items[SavedQue	else
ryCombo.ItemIndex];	begin
QueryEdit.Text :=	<b>B</b> ^:
IniStrToStr(ReadString(QueryNam	Inc(
e, 'Query', ''));	Inc(
AliasCombo.Text :=	end;
ReadString(QueryName, 'Alias', '');	B^ := 7
NameEdit.Text :=	Result
ReadString(QueryName, 'Name', '');	end;
end;	
Unmodify;	function
if Showing then	Boolean
if NameEdit.Text <> '' then	begin
PasswordEdit.SetFocus else	Result
QueryEdit.SetFocus;	OldAlia
end;	Quer
	end;
<> procedure	
TAdhocForm.SaveQueryAs;	function
begin	TAdhoc
if GetNewName(QueryName) then	Boolean
begin	begin
Unnamed := False;	Result
SaveQuery;	if IsMo
with SavedQueryCombo, Items do	case I
begin	%s mod
if IndexOf(QueryName) < 0 then	mtC
Add(QueryName);	mbYesN
ItemIndex :=	mrY
IndexOf(QueryName);	mrC
end;	end;
end;	end;
end;	
	<>
<> procedure	TAdhoc
TAdhocForm.SaveQuery;	begin
begin	OldAli
if Unnamed then	Namel
SaveQueryAs	<u> </u>

while S^ <> #0 do
if $(S[0] = '\')$ and $(S[1] = 'n')$ then
begin
B^ := #13;
Inc(B);
B^ := #10;
Inc(B);
Inc(S);
Inc(S);
end
else
begin
B^ := S^;
Inc(B);
Inc(S);
end;
B^ := #0;
Result := Buffer;
end;
,
function TAdhocForm.IsModified:
Boolean;
begin
Result := (AliasCombo.Text <>
OldAlias) or NameEdit.Modified or
QueryEdit.Modified;
end;
cnu,
function
TAdhocForm.CheckModified:
Boolean;
begin
Result := True;
if IsModified then
case MessageDlg(Format('Query
%s modified, save?', [QueryName]),
mtConfirmation,
mbYesNoCancel, 0) of
mrYes: SaveQuery;
mrCancel: Result := False;
end;
end;
,
<> procedure
TAdhocForm.Unmodify;
begin
OldAlias := AliasCombo.Text;
,
NameEdit.Modified := False;

'Alias', 'IBLOCAL');	else
Allas, IDLOCAL J,	with SavedQueries do
Write String (Very Inefficient Name	
WriteString(VeryInefficientName,	begin
'Name', 'SYSDBA');	WriteString(QueryName,
	'Query',
SavedQueryCombo.Items.Add(Very	StrToIniStr(QueryEdit.Text));
InefficientName);	WriteString(QueryName, 'Alias',
WriteString(AmountDueName,	AliasCombo.Text);
'Query', AmountDueByCustomer);	WriteString(QueryName,
WriteString(AmountDueName,	'Name', NameEdit.Text);
'Alias', 'DBDEMOS');	Unmodify;
WriteString(AmountDueName,	end;
'Name', '');	end;
SavedQueryCombo.Items.Add(Amo	<> procedure
untDueName);	TAdhocForm.FormCreate(Sender:
end;	TObject);
end;	<> procedure
	CreateInitialIni;
	const
begin	VeryInefficientName = 'IB: Very
{ Grab session aliases }	Inefficient Query';
	VeryInefficientQuery =
Session.GetAliasNames(AliasCombo	'select EMP NO, Avg(Salary) as
.Items);	Salary\n'+
	' from employee, employee,
{ Load in saved queries }	employee\n' +
SavedQueries :=	' group by EMP NO';
TIniFile.Create('BKQUERY.INI');	AmountDueName = 'DB: Amount
77	Due By Customer';
SavedQueries.ReadSections(SavedQ	AmountDueByCustomer =
ueryCombo.Items);	'select Company,
if SavedQueryCombo.Items.Count	Sum(ItemsTotal) -
<= 0 then CreateInitialIni;	Sum(AmountPaid) as AmountDue\n'
SavedQueryCombo.ItemIndex := 0;	+
QueryName :=	' from customer, orders\n' +
SavedQueryCombo.Items[0];	' where Customer.CustNo =
Unmodify;	Orders.CustNo\n' +
ReadQuery;	' group by Company';
end;	begin
	{ Create initial INI file when one
<> procedure	doesn't already exisit }
TAdhocForm.FormDestroy(Sender:	with SavedQueries do
TObject);	begin
begin	begin
SavedQueries.Free;	WriteString(VeryInefficientName,
end;	'Query', VeryInefficientQuery);
Ciru,	Query, verymemicientQuery);
/ > procedure	WriteString(VeryInefficientNems
<> procedure	WriteString(VeryInefficientName,

TObject);	TA
begin	: T(
SaveQuery;	beg
end;	Cl
	end
<> procedure	
TAdhocForm.SaveAsBtnClick(Send	<
er: TObject);	TA
begin	er:
SaveQueryAs;	beg
end;	Ba
,	Alia
<> procedure	Pas
TAdhocForm.SavedQueryComboCh	
ange(Sender: TObject);	Br
begin	end
ReadQuery;	
end;	<
	TA
<> procedure	TO
TAdhocForm.FormCloseQuery(Sen	
der: TObject;	fu
var CanClose: Boolean);	va
begin	I
CanClose := CheckModified;	be
end;	I
	r
end.	
	[I]):
111111111111111111111111111111111111111	u
2	Sav
unit ResltFrm;	Res
	en
interface	
	beg
uses	Al
Windows, Messages, SysUtils,	Na
Classes, Graphics, Controls, Forms,	Pa
Dialogs,	Qı
ExtCtrls, DB, DBTables, Grids,	Qı
DBGrids, StdCtrls;	Qı
	Sa
type	1;
TQueryForm = class(TForm)	Ur
QueryLabel: TLabel;	end
DBGrid1: TDBGrid;	1
DataSource: TDataSource;	<
	1

TAdhocForm.CloseBtnClick(Sender
: TObject);
begin
Close;
end;
<> procedure
TAdhocForm.ExecuteBtnClick(Send
er: TObject);
begin
BackgroundQuery(QueryName,
AliasCombo.Text, NameEdit.Text,
PasswordEdit.Text,
QueryEdit.Text);
BringToFront;
end;
<> procedure
TAdhocForm.NewBtnClick(Sender:
TObject);
1 object);
function UniqueName: string;
var
I: Integer;
begin
I := 1;
· · · · · · · · · · · · · · · · · · ·
repeat  Result := Format('Query%d',
[I]);
Inc(I);
until
SavedQueryCombo.Items.IndexOf( Result) < 0;
end;
hagin
begin
AliasCombo.Text := 'DBDEMOS';
NameEdit.Text := ";
PasswordEdit.Text := ";
QueryEdit.Text := ";
QueryEdit.SetFocus;
QueryName := UniqueName;
SavedQueryCombo.ItemIndex := -
1;
Unnamed := True;
end;
<> procedure
TAdhocForm.SaveBtnClick(Sender:

Guard: Integer;
Numbers: Integer;
{ Thread safe increment of Numbers
to guarantee the result is unique }
function GetUniqueNumber:
Integer;
asm
@@1: MOV EDX,1
XCHG Guard,EDX
OR EDX,EDX
,
JNZ @@2
MOV EAX, Numbers
INC EAX
MOV Numbers,EAX
MOV Guard,EDX
RET
OOA BUGU A
@@2: PUSH 0
CALL Sleep
JMP @@1
end;
<> procedure
TQueryThread.Execute;
var
UniqueNumber: Integer;
begin
try
with QueryForm do
begin
{ Ensure the Query has a unique
session and database. A unique
session
is required for each thread.
Since databases are session specific
it must be unique as well }
UniqueNumber :=
GetUniqueNumber;
Session.SessionName :=
Format('%s%x', [Session.Name,
UniqueNumber]);
Database.SessionName :=
Session.SessionName;
,
Database.DatabaseName :=
Format('%s%x', [Database.Name,
TOTHICL /05/0A , Databascitaine,

0 70
Query: TQuery;
Session: TSession;
StatusLine: TLabel;
Database: TDatabase;
private
{ Private declarations }
public
{ Public declarations }
end;
- Chu,
<> procedure
<u> </u>
BackgroundQuery(const
QueryName, Alias, User, Password,
QueryText: string);
implementation
{\$R *.dfm}
{ TQueryThread }
( 2 ( 100 ) 2 12 2 100 2 )
type
TQueryThread = class(TThread)
private
QueryForm: TQueryForm;
MessageText: string;
<> procedure
ConnectQuery;
<> procedure
DisplayMessage;
protected
<> procedure Execute;
override;
public
constructor Create(AQueryForm:
TQueryForm);
•
end;
constructor
TQueryThread.Create(AQueryFor
m: TQueryForm);
begin
QueryForm := AQueryForm;
FreeOnTerminate := True;
inherited Create(False);
end;
,
var
7 661

with QueryForm do	UniqueNumber]);
StatusLine.Caption := MessageText;	Query.SessionName :=
end;	Database.SessionName;
,	Query.DatabaseName :=
{ BackgroundQuery }	Database.DatabaseName;
<> procedure	{ Open the query }
BackgroundQuery(const	Query.Open;
	Query. Open,
QueryName, Alias, User, Password,	
QueryText: string);	{ Connect the query to the grid.
var	This must be done in a synchronzied
QueryForm: TQueryForm;	method since assigning the
begin	query to the DataSource will modify
QueryForm :=	the
TQueryForm.Create(Application);	contents of the grid and the grid
with QueryForm, Database do	can only be modified from the main
begin	VCL thread }
Caption := QueryName;	Synchronize(ConnectQuery);
QueryLabel.Caption :=	
QueryText;	{ Update the status line. Since
Show;	the label is a VCL control this must
AliasName := Alias;	also be done in the main VCL
	thread }
Params.Values['USER'] := User;	MessageText := 'Query
Params.Values['PASSWORD'] :=	openned';
Password;	
Query.Sql.Text := QueryText;	Synchronize(DisplayMessage);
end;	end;
	except
{ Create the background thread to	on E: Exception do
execute the query. Since the thread	begin
will free itself on termination we	{ Display any error we receive on
do not neet to maintain a reference	the status line }
to it. Creating it is enough }	MessageText := Format('%s:
	%s.', [E.ClassName, E.Message]);
TQueryThread.Create(QueryForm);	Synchronize(DisplayMessage);
end;	end;
	end;
end.	end;
Circ.	
\\\\\\\\\\	<> procedure
	TQueryThread.ConnectQuery;
3	
unit SaveQAs;	begin
	with QueryForm do
interface	DataSource.Dataset := Query;
	end;
uses	
Windows, Messages, SysUtils,	<> procedure
Classes, Graphics, Controls, Forms,	TQueryThread.DisplayMessage;
Dialogs,	begin
<b>G</b> /	

# القسم السادس برمجة الألعاب في لغة البرمجة دلفي

الهاف التنفيذي لبرنامج
24 procedures
Swat)
إعداد
علاء الدين اللباد
واتف ۷۳۷۱۹۰۱
unit Main;
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
ExtCtrls, Menus, StdCtrls;
const
crMaletUp : integer = 5;
crMaletDown : integer = 6;
MissedPoints : integer = -2;
HitPoints: integer = 5;
MissedCritter : integer = -1;
CritterSize : integer = 72;
TimerId: integer = 1;
type
THole = record
Time : integer;

StdCtrls, ExtCtrls;
type
TSaveQueryAs = class(TForm)
NameEdit: TEdit;
Label1: TLabel;
Bevel1: TBevel;
OKBtn: TButton;
CancelBtn: TButton;
<> procedure
NameEditChange(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
function GetNewName(var
QueryName: string): Boolean;
<b>9</b> /
implementation
Implementation
(CD * dfm)
{\$R *.dfm}
C (N N)
function GetNewName(var
QueryName: string): Boolean;
begin
with
TSaveQueryAs.Create(Application)
do
begin
NameEdit.Text := QueryName;
Result := ShowModal = mrOK;
if Result then QueryName :=
NameEdit.Text;
end;
end;
cnu,
<pre>&lt;&gt; procedure TS On and A Name E EACh (See </pre>
TSaveQueryAs.NameEditChange(Se
nder: TObject);
begin
OkBtn.Enabled := NameEdit.Text
⟨> ";
end;
end.
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

Dead : TBitmap;
HoleInfo: array[04] of THole;
Holes: array[04] of TPoint;
<> procedure WriteScore;
public
{ Public declarations }
LiveTime, Frequence, GameTime
: integer;
end;
var
SwatForm: TSwatForm;
Swatt of m. 15 watt of m,
implementation
Implementation
uses options, about;
uses options, about,
{\$R *.dfm}
-
{\$R extrares.res}
<> procedure
TSwatForm.FormCreate(Sender:
TObject);
begin
Holes[0] := Point(10, 10);
Holes[1] := Point( 200, 10 );
Holes[2] := Point( 100, 100 );
Holes[3] := Point( 10, 200 );
Holes[4] := Point( 200, 200 );
Screen.Cursors[crMaletUp] :=
LoadCursor(HInstance, 'Malet');
Screen.Cursors[crMaletDown] :=
LoadCursor(HInstance,
'MaletDown');
Screen.Cursor :=
TCursor(crMaletUp);
-
randomize;
Live := TBitmap.Create;
Live.LoadFromResourceName(HIns
tance, 'Live');
Dead := TBitmap.Create;
Dead : 15thap.Cicate,
Dead.LoadFromResourceName(HIn
stance, 'Dead');
stance, Deau J,

Dead : boolean; end;  TSwatForm = class(TForm) MainMenu1: TMainMenu; Gamr1: TMenuItem;
TSwatForm = class(TForm) MainMenu1: TMainMenu;
MainMenu1: TMainMenu;
MainMenu1: TMainMenu;
· ·
Gamrl: TMenultem:
,
New1: TMenuItem;
Options1: TMenuItem;
Stop1: TMenuItem;
Pause1: TMenuItem;
About1: TMenuItem;
Timer1: TTimer;
GameOverImage: TImage;
Image1: TImage;
TimeLabel: TLabel;
MissLabel: TLabel;
HitsLabel: TLabel;
EscapedLabel: TLabel;
ScoreLabel: TLabel;
<> procedure
FormCreate(Sender: TObject);
<> procedure
Timer1Timer(Sender: TObject);
<> procedure
FormMouseDown(Sender: TObject;
<b>Button: TMouseButton;</b>
Shift: TShiftState; X, Y:
Integer);
<> procedure
FormMouseUp(Sender: TObject;
Button: TMouseButton;
Shift: TShiftState; X, Y:
Integer);
<> procedure
New1Click(Sender: TObject);
<> procedure
Options1Click(Sender: TObject);
<> procedure
Stop1Click(Sender: TObject);
<> procedure
Pause1Click(Sender: TObject);
<> procedure
About1Click(Sender: TObject);
private
{ Private declarations }
Score : integer;
Hits, Miss, Escaped : integer;
IsGameOver, IsPause : Boolean;
Live: TBitmap;

WriteScore;	IsGameOver := true;
if (Timer1.Tag >= GameTime) then	IsPause := false;
Stop1Click(self);	LiveTime := 10;
end;	Frequence := 20;
,	GameTime := 150;
<> procedure	seconds
TSwatForm.FormMouseDown(Send	
er: TObject; Button:	Application.OnMinim
TMouseButton; Shift: TShiftState;	Pause1Click;
X, Y: Integer);	Application.OnRestor
var	Pause1Click;
i : integer;	end;
hit : boolean;	
begin	<> procedure
Screen.Cursor :=	TSwatForm.Timer1Ti
TCursor(crMaletDown);	TObject);
	var
if (IsGameOver or IsPause) then	i : integer;
exit;	begin
	Timer1.Tag := Timer
hit := false;	i := random(Frequenc
for i := 0 to 4 do	if (i < 5) then
if ( (not HoleInfo[i].Dead) and	begin
(HoleInfo[i].Time $\Leftrightarrow$ 0) ) then	if (HoleInfo[i].Time
if (X > Holes[i].x ) and ( X <	begin
(Holes[i].x + Live.Width)) and	HoleInfo[i].Time :=
(Y > Holes[i].y) and $(Y <$	+ LiveTime;
(Holes[i].y + Live.Height)) then	HoleInfo[i].Dead :=
begin	Canvas.Draw(Holes
<pre>inc( Score, HitPoints );</pre>	Holes[i].y, Live);
HoleInfo[i].Dead := true;	end;
HoleInfo[i].Time := Timer1.Tag	end;
+ 2 * LiveTime;	for i := 0 to 4 do
inc( Hits );	begin
hit := true;	if ( (Timer1.Tag >
Canvas.Draw(Holes[i].x,	HoleInfo[i].Time ) and
Holes[i].y, Dead);	HoleInfo[i].Time <> 0
end;	begin
if not(hit) then	HoleInfo[i].Time :=
begin	if not(HoleInfo[i].D
inc (Score, MissedPoints);	begin
inc( Miss );	inc( Score, Missed
end;	inc( Escaped );
WriteScore;	end;
end;	Canvas.FillRect(Re
	Holes[i].y, Holes[i].x +
<> procedure	Holes   .y + Deau.Heigh
<> procedure TSwatForm.FormMouseUp(Sender:	Holes[i].y + Dead.Heiglend;

IsGameOver := true;
IsPause := false;
LiveTime := 10;
Frequence := 20;
GameTime := 150; // fifteen
seconds
Application.OnMinimize :=
Pause1Click;
Application.OnRestore :=
Pause1Click;
end;
,
<> procedure
TSwatForm.Timer1Timer(Sender:
TObject);
var
i: integer;
begin
Timer1.Tag := Timer1.Tag + 1;
i := random(Frequence);
if (i < 5) then
begin
if (HoleInfo[i].Time = 0) then
begin
HoleInfo[i].Time := Timer1.Tag
+ LiveTime;
HoleInfo[i].Dead := false;
Canvas.Draw(Holes[i].x,
Holes[i].y, Live);
end;
end;
for i := 0 to 4 do
begin
if ( (Timer1.Tag >
HoleInfo[i].Time ) and (
HoleInfo[i].Time <> 0 ) ) then
begin
HoleInfo[i].Time := 0;
if not(HoleInfo[i].Dead) then
begin
inc( Score, MissedCritter );
inc( Escaped );
end;
Canvas.FillRect(Rect(Holes[i].x,
Holes[i].y, Holes[i].x + Dead.Width,
Holes[i].y + Dead.Height));
end;
end;
CIIU,

New1.Enabled := true;
Options1.Enabled := true;
Stop1.Enabled := false;
for i := 0 to 4 do
if (HoleInfo[i].Time $\Leftrightarrow$ 0) then
Canvas.FillRect(Rect(Holes[i].x,
Holes[i].y, Holes[i].x + Dead.Width,
Holes[i].y + Dead.Height));
end;
<> procedure
TSwatForm.Pause1Click(Sender:
TObject);
begin
if (IsGameOver) then
exit;
,
if (IsPause) then
begin
IsPause := false;
Pause1.Caption := '&Pause';
Stop1.Enabled := true;
Timer1.Enabled := true;
end
else
begin
IsPause := true;
Pause1.Caption := '&Continue';
Stop1.Enabled := false;
Timer1.Enabled := false;
end:
/
end;
<> procedure
TSwatForm.About1Click(Sender:
TObject);
begin
AboutBox.ShowModal;
end;
CHU,
Z Nuccedune
<> procedure TSwatEown WhiteScores
TSwatForm.WriteScore;
begin T: I I I C 4:
TimeLabel.Caption :=
IntToStr(GameTime - Timer1.Tag);
HitsLabel.Caption :=
IntToStr(Hits);
MissLabel.Caption :=

Shift: TShiftState; X, Y: Integer);
begin
Screen.Cursor :=
TCursor(crMaletUp);
end;
cnu,
<> procedure
TSwatForm.New1Click(Sender:
TObject);
begin
Timer1.Enabled := true;
Timer1.Tag := 0;
Score := 0;
Hits := 0;
Miss := 0;
Escaped := 0;
if (IsPause)
then begin
IsPause := false;
Pause1.Caption := '&Pause';
end;
GameOverImage.Visible := false;
IsGameOver := false;
FillChar(HoleInfo,
sizeof(HoleInfo), 0);
New1.Enabled := false;
Options1.Enabled := false;
-
Stop1.Enabled := true;
end;
<> procedure
TSwatForm.Options1Click(Sender:
TObject);
begin
OptionsDlg.ShowModal;
end;
,
<> procedure
TSwatForm.Stop1Click(Sender:
<u> </u>
TObject);
var
i : integer;
begin
Timer1.Enabled := false;
IsPause := false;
GameOverImage.Visible := true;
IsGameOver := true;
Timer1.Tag := GameTime;
Time 1.1 ag Game I mie,

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	
3	
unit options;	
interface	
interrace	
uses	
Windows, Messages, SysUtils,	
Classes, Graphics, Controls, Forms,	
Dialogs,	
StdCtrls, ComCtrls, ExtCtrls;	
type	
TOptionsDlg = class(TForm)	
Bevel1: TBevel;	
Slow: TLabel;	
Fast: TLabel;	
Label1: TLabel;	
Label2: TLabel;	
Speed: TLabel;	
Population: TLabel;	
Time: TLabel;	
OKBtn: TButton;	
CancelBtn: TButton;	
SpeedSet: TTrackBar;	
PopulationSet: TTrackBar;	
GameTimeSet: TEdit;	
<> procedure	
OKBtnClick(Sender: TObject);	
<> procedure	
FormShow(Sender: TObject);	
private	
{ Private declarations }	
public	
{ Public declarations }	
end;	
var	
OptionsDlg: TOptionsDlg;	
implementation	
uses Main;	
uses main,	
{\$R *.dfm}	
<> procedure	
TOptionsDlg.OKBtnClick(Sender:	

IntToStr(Miss);
EscapedLabel.Caption :=
IntToStr(Escaped);
ScoreLabel.Caption :=
IntToStr(Score);
end;
end.
ciu.
\\\\\\\\\
2
unit about;
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
StdCtrls, ExtCtrls;
2000 2000 2000 2000
type
TAboutBox = class(TForm)
Panel1: TPanel;
ŕ
ProgramIcon: TImage;
ProductName: TLabel;
Hits: TLabel;
Copyright: TLabel;
Comments: TLabel;
Escape: TLabel;
Miss: TLabel;
Label1: TLabel;
OKButton: TButton;
private
{ Private declarations }
public
{ Public declarations }
end;
cnu,
var
AboutBox: TAboutBox;
implementation
{\$R *.dfm}
end.
ļ

TObject);

Degin
SwatForm.LiveTime :=
SpeedSet.Max + 1 -
SpeedSet.Position;
SwatForm.Frequence :=
PopulationSet.Position;
SwatForm.GameTime :=
StrToInt(GameTimeSet.Text);
// limit the value of GameTime to a
reasonable length
if (SwatForm.GameTime < 1) then
SwatForm.GameTime := 150;
if (SwatForm.GameTime > 9999)
then
SwatForm.GameTime := 9999;
end;
<> procedure
TOptionsDlg.FormShow(Sender:
TObject);
begin
SpeedSet.Position := SpeedSet.Max
+ 1 - SwatForm.LiveTime;
PopulationSet.Position :=
SwatForm.Frequence;
GameTimeSet.Text :=
inttoStr(SwatForm.GameTime);
end;
end.
\\\\\\\\\\\\

<> object \$41	
الهاف النصي لبرناهج	
Swat)	
إعداد	
علاء الدين اللباد	
واتف ۷۳۳۵۷۵۹۹۰ واتف ۱۷۳۳۵۷۵۹۹۰	
<> object SwatForm:	_
TSwatForm	
Left = 570	
Top = 159	
Width = 308	
Height = 494	
Caption = 'Swat!'	
Color = clSilver	
Font.Charset =	
DEFAULT CHARSET	
Font.Color = clWindowText	_
Font.Height = -11	
Font.Name = 'MS Sans Serif'	_
Font.Style = []	
Menu = MainMenu1	
OldCreateOrder = False	
OnCreate = FormCreate	_
OnMouseDown =	_
FormMouseDown	

Top = 351
Width = 9
Height = 19
Caption = '0'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -16
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
Transparent = True
end
<> object EscapedLabel:
TLabel
Left = 134
Top = 351
Width = 9
Height = 19
Caption = '0'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -16
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
Transparent = True
end
<> object ScoreLabel: TLabel
Left = 205
Top = 351
Width = 9
Height = 19
Caption = '0'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -16
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
Transparent = True
end
<> object MainMenu1:
TMainMenu
<> object Gamr1:
TMenuItem
11/10/14/10/11

OnMouseUp = FormMouseUp
PixelsPerInch = 96
TextHeight = 13
<> object GameOverImage:
TImage
Left = 10
Top = 72
Width = 262
Height = 145
AutoSize = True
Picture.Data = {
88888888888888888888888888888888888888
Transparent = True
end
<> object TimeLabel: TLabel
Left = 122
Top = 296
Width = 18
Height = 19
Caption = '00'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -16
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
Transparent = True
end
<> object MissLabel: TLabel
Left = 78
Top = 351
Width = 9
Height = 19
Caption = '0'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -16
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
Transparent = True
end
<> object HitsLabel: TLabel
Left = 41

Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = [] OldCreateOrder = False
Position = poScreenCenter
OnShow = FormShow
PixelsPerInch = 96
TextHeight = 13
<> object Bevel1: TBevel
Left = 8
Top = 8
Width = 281
Height = 217
Shape = bsFrame
end
<> object Slow: TLabel
Left = 36
Top = 72
Width = 23
Height = 13
Caption = 'Slow'
end
<> object Fast: TLabel
Left = 241
Top = 72
Width = 20
Height = 13
Caption = 'Fast'
end
<> object Label1: TLabel
Left = 36
Top = 144
Width = 20
Height = 13
Caption = 'Low'
end
<> object Label2: TLabel
Left = 239
Top = 144
Width = 22
Height = 13
Caption = 'High'
end
<> object Speed: TLabel
Left = 133

Caption = '&Game'
<> object New1:
TMenuItem
Caption = '&New'
OnClick = New1Click
end
<> object Options1:
TMenuItem
Caption = '&Options'
OnClick = Options1Click
end
<> object Stop1:
TMenuItem
Caption = '&Stop'
Enabled = False
OnClick = Stop1Click
end
end
<> object Pause1:
TMenuItem
Caption = '&Pause'
OnClick = Pause1Click
end
<> object About1:
TMenuItem
Caption = '&About'
OnClick = About1Click
end
end
<> object Timer1: TTimer
Enabled = False
Interval = 100
OnTimer = Timer1Timer
Left = 32
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
2
<> object OptionsDlg:
TOptionsDlg
Left = 174
Top = 188
BorderStyle = bsDialog
Caption = 'Options'
ClientHeight = 233
ClientWidth = 384
Color = clBtnFace
Coloi – Ciduir ace

Height = 25
Max = 30
TabOrder = 2
end
<> object PopulationSet:
TTrackBar
Left = 36
Top = 112
Width = 225
Height = 25
Max = 35
TabOrder = 3
end
<> object GameTimeSet:
TEdit
Left = 116
Top = 184
Width = 65
Height = 21
TabOrder = 4
Text = '150'
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
3
<> object AboutBox:
TAboutBox
Left = 150
Top = 135
BorderStyle = bsDialog
Caption = 'About Swat!'
ClientHeight = 213
ClientWidth = 209
Color = clBtnFace
Font.Charset =
1 one. Charget
DEFAULT_CHARSET
DEFAULT_CHARSET
DEFAULT_CHARSET Font.Color = clWindowText
DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []
DEFAULT_CHARSET Font.Color = clWindowText Font.Height = -11 Font.Name = 'MS Sans Serif'
DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []
DEFAULT_CHARSET Font.Color = clWindowText Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = False
DEFAULT_CHARSET Font.Color = clWindowText Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = False Position = poScreenCenter PixelsPerInch = 96 TextHeight = 13
DEFAULT_CHARSET Font.Color = clWindowText Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = False Position = poScreenCenter PixelsPerInch = 96 TextHeight = 13 <> object Panel1: TPanel
DEFAULT_CHARSET Font.Color = clWindowText Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = False Position = poScreenCenter PixelsPerInch = 96 TextHeight = 13 <> object Panel1: TPanel Left = 8
DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  OldCreateOrder = False  Position = poScreenCenter  PixelsPerInch = 96  TextHeight = 13  <> object Panel1: TPanel

Top = 24
Width = 31
Height = 13
Caption = '&Speed'
FocusControl = SpeedSet
end
<> object Population: TLabel
Left = 123
Top = 96
Width = 50
Height = 13
Caption = '&Population'
FocusControl = PopulationSet
end
<> object Time: TLabel
Left = 121
Top = 168
Width = 54
Height = 13
Caption = '&Game Time'
FocusControl = GameTimeSet
end
<> object OKBtn: TButton
Left = 300
Top = 8
Width = 75
Height = 25
Caption = 'OK'
Default = True
ModalResult = 1
TabOrder = 0
OnClick = OKBtnClick
end
<> object CancelBtn: TButton
Left = 300
Top = 38
Width = 75
Height = 25
Cancel = True
Caption = 'Cancel'
ModalResult = 2
TabOrder = 1
end
<> object SpeedSet:
TTrackBar
Left = 36
Top = 40
Width = 225

Top = 104 Width = 73 Height = 13 Alignment = taCenter Caption = 'Delphi Example' WordWrap = True IsControl = True end <> object Escape: TLabel Left = 55 Top = 72 Width = 82 Height = 13 Alignment = taCenter Caption = '-1 pts per escape' IsControl = True end <> object Miss: TLabel Left = 63 Top = 88 Width = 67 Height = 13 Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1 end	I of 4 - 41
Width = 73 Height = 13 Alignment = taCenter Caption = 'Delphi Example' WordWrap = True IsControl = True end <> object Escape: TLabel Left = 55 Top = 72 Width = 82 Height = 13 Alignment = taCenter Caption = '-1 pts per escape' IsControl = True end <> object Miss: TLabel Left = 63 Top = 88 Width = 67 Height = 13 Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	Left = 61
Height = 13 Alignment = taCenter Caption = 'Delphi Example' WordWrap = True IsControl = True end <> object Escape: TLabel Left = 55 Top = 72 Width = 82 Height = 13 Alignment = taCenter Caption = '-1 pts per escape' IsControl = True end <> object Miss: TLabel Left = 63 Top = 88 Width = 67 Height = 13 Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
Alignment = taCenter Caption = 'Delphi Example' WordWrap = True IsControl = True end <> object Escape: TLabel Left = 55 Top = 72 Width = 82 Height = 13 Alignment = taCenter Caption = '-1 pts per escape' IsControl = True end <> object Miss: TLabel Left = 63 Top = 88 Width = 67 Height = 13 Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'borland International' IsControl = True end end> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
Caption = 'Delphi Example' WordWrap = True IsControl = True end <> object Escape: TLabel Left = 55 Top = 72 Width = 82 Height = 13 Alignment = taCenter Caption = '-1 pts per escape' IsControl = True end <> object Miss: TLabel Left = 63 Top = 88 Width = 67 Height = 13 Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
WordWrap = True IsControl = True end <> object Escape: TLabel Left = 55 Top = 72 Width = 82 Height = 13 Alignment = taCenter Caption = '-1 pts per escape' IsControl = True end <> object Miss: TLabel Left = 63 Top = 88 Width = 67 Height = 13 Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	<u> </u>
IsControl = True end <> object Escape: TLabel Left = 55 Top = 72 Width = 82 Height = 13 Alignment = taCenter Caption = '-1 pts per escape' IsControl = True end <> object Miss: TLabel Left = 63 Top = 88 Width = 67 Height = 13 Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
end <> object Escape: TLabel Left = 55 Top = 72 Width = 82 Height = 13 Alignment = taCenter Caption = '-1 pts per escape' IsControl = True end <> object Miss: TLabel Left = 63 Top = 88 Width = 67 Height = 13 Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
<pre>&lt;&gt; object Escape: TLabel Left = 55 Top = 72 Width = 82 Height = 13 Alignment = taCenter Caption = '-1 pts per escape' IsControl = True end &lt;&gt; object Miss: TLabel Left = 63 Top = 88 Width = 67 Height = 13 Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end &lt;&gt; object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end &lt;&gt; object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1</pre>	
Left = 55  Top = 72  Width = 82  Height = 13  Alignment = taCenter  Caption = '-1 pts per escape' IsControl = True end > object Miss: TLabel  Left = 63  Top = 88  Width = 67  Height = 13  Alignment = taCenter  Caption = '-2 pts per miss' IsControl = True end > object Label1: TLabel  Left = 48  Top = 136  Width = 97  Height = 13  Alignment = taCenter  Caption = 'Borland  International' IsControl = True end end > object OKButton: TButton  Left = 67  Top = 180  Width = 75  Height = 25  Caption = 'OK' Default = True  ModalResult = 1  TabOrder = 1	
Top = 72 Width = 82 Height = 13 Alignment = taCenter Caption = '-1 pts per escape' IsControl = True end <> object Miss: TLabel Left = 63 Top = 88 Width = 67 Height = 13 Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	<> object Escape: TLabel
Width = 82 Height = 13 Alignment = taCenter Caption = '-1 pts per escape' IsControl = True end <> object Miss: TLabel Left = 63 Top = 88 Width = 67 Height = 13 Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
Height = 13 Alignment = taCenter Caption = '-1 pts per escape' IsControl = True end <> object Miss: TLabel Left = 63 Top = 88 Width = 67 Height = 13 Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
Alignment = taCenter Caption = '-1 pts per escape' IsControl = True end <> object Miss: TLabel Left = 63 Top = 88 Width = 67 Height = 13 Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
Caption = '-1 pts per escape'  IsControl = True end  <> object Miss: TLabel  Left = 63  Top = 88  Width = 67  Height = 13  Alignment = taCenter  Caption = '-2 pts per miss'  IsControl = True end  <> object Label1: TLabel  Left = 48  Top = 136  Width = 97  Height = 13  Alignment = taCenter  Caption = 'Borland  International'  IsControl = True end end  <> object OKButton: TButton  Left = 67  Top = 180  Width = 75  Height = 25  Caption = 'OK'  Default = True  ModalResult = 1  TabOrder = 1	Height = 13
IsControl = True end <> object Miss: TLabel Left = 63 Top = 88 Width = 67 Height = 13 Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	Alignment = taCenter
IsControl = True end <> object Miss: TLabel Left = 63 Top = 88 Width = 67 Height = 13 Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	Caption = '-1 pts per escape'
end <> object Miss: TLabel  Left = 63  Top = 88  Width = 67  Height = 13  Alignment = taCenter Caption = '-2 pts per miss'  IsControl = True end <> object Label1: TLabel Left = 48  Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67  Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
Left = 63 Top = 88 Width = 67 Height = 13 Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	end
Left = 63 Top = 88 Width = 67 Height = 13 Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	<> object Miss: TLabel
Width = 67 Height = 13 Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
Width = 67 Height = 13 Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	Top = 88
Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
Alignment = taCenter Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	Height = 13
Caption = '-2 pts per miss' IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
IsControl = True end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
end <> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
<> object Label1: TLabel Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
Left = 48 Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
Top = 136 Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	9
Width = 97 Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
Height = 13 Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	•
Alignment = taCenter Caption = 'Borland International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
Caption = 'Borland International'  IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
International' IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	Š.
IsControl = True end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	<u> </u>
end end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
end <> object OKButton: TButton Left = 67 Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
<> object OKButton: TButton  Left = 67  Top = 180  Width = 75  Height = 25  Caption = 'OK'  Default = True  ModalResult = 1  TabOrder = 1	7 22 32
Left = 67  Top = 180  Width = 75  Height = 25  Caption = 'OK'  Default = True  ModalResult = 1  TabOrder = 1	
Top = 180 Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	9
Width = 75 Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
Height = 25 Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	•
Caption = 'OK' Default = True ModalResult = 1 TabOrder = 1	
Default = True ModalResult = 1 TabOrder = 1	Height = 25
ModalResult = 1 TabOrder = 1	
TabOrder = 1	
end	
	end

Width = 193
Height = 161
BevelInner = bvRaised
BevelOuter = bvLowered
Caption = 'Panel1'
ParentColor = True
TabOrder = 0
<> object ProgramIcon:
TImage
Left = 80
Top = 24
Width = 33
Height = 33
Picture.Data = {
Forces:
E0FC0003}
Stretch = True
IsControl = True
end
<> object ProductName:
TLabel
Left = 83
Top = 8
<b>Width = 27</b>
Height = 13
Caption = 'Swat!'
IsControl = True
end
<> object Hits: TLabel
Left = 69
Top = 56
Width = 55
Height = 13
Alignment = taCenter
Caption = '5 pts per hit'
IsControl = True
end
<> object Copyright: TLabel
Left = 55
Top = 120
<b>Width</b> = <b>83</b>
Height = 13
Alignment = taCenter
Caption = 'Copyright © 1998'
IsControl = True
end
<> object Comments:
TLabel

Shift: TShiftState; X, Y: Integer);
<> procedure
FormCreate(Sender: TObject);
private
{ Private declarations }
FGameOver : Boolean;
public
{ Public declarations }
backgroundImage : TImage;
spriteImage : TImage;
paddle : TImage;
backgroundCanvas : TCanvas;
workCanvas : TCanvas;
backgroundRect, spriteRect,
changeRect, paddleRect,
changePaddleRect :TRect;
x, y, xDir, yDir, paddleX,
paddleY, paddleCenter, Angle :
integer;
<> procedure IdleLoop(
Sender: TObject; var Done: Boolean
);
<> procedure
WMSetCursor(var Message:
TWMSetCursor); message
WM_SETCURSOR;
end;
var
Form1: TForm1;
implementation
{\$R *.dfm}
uses MMSystem;
•
<> procedure
TForm1.FormPaint(Sender:
TObject);
begin
_ ~ - 8
RealizePalette(backgroundCanvas.H
andle);
RealizePalette(workCanvas.Handle);
realizer arctic work canvastrandic),
Canvas.CopyRect(backgroundRect,
workCanvas, backgroundRect);
workanyas, vackgroundikectj,

End	
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	

الملف التنفيذي لبرنامج
12 procedures
EarthPng)
عامدا
علاء الدين اللباد
واتف ۷۹۲۵۷۵۲۹۰
unit Main;
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs, ExtCtrls;
Arvas
type TForm1 = class(TForm)
<> procedure
FormPaint(Sender: TObject);
<> procedure
FormActivate(Sender: TObject);
<> procedure
FormMouseMove(Sender: TObject;

CreateCompatibleDC(Canvas.Handl	end;
e);	
bkbmp :=	<> procedure
CreateCompatibleBitmap(Canvas.H	TForm1.FormActivate(Sender:
andle, ClientWidth, ClientHeight);	TObject);
SelectObject(backgrounddc,	var
bkbmp);	backgrounddc, workdc : HDC;
SelectPalette(backgrounddc,	bkbmp, bmp : HBITMAP;
backgroundImage.Picture.Bitmap.P	begin
alette, false);	backgroundImage :=
backgroundCanvas.Handle :=	TImage.Create(Self);
backgrounddc;	spriteImage := TImage.Create( Self
backgroundCanvas.StretchDraw(	):
backgroundRect,	paddle := TImage.Create( Self );
backgroundImage.Picture.Bitmap);	workCanvas := TCanvas.Create;
	backgroundCanvas :=
	TCanvas.Create;
//Set up workCanvas	
workdc :=	Angle := 1;
CreateCompatibleDC(Canvas.Handl	ingre ( 1)
e);	
bmp :=	spriteImage.Picture.LoadFromFile('
CreateCompatibleBitmap(Canvas.H	Earth.ico');
andle, ClientWidth, ClientHeight);	zurenico),
SelectObject(workdc, bmp);	backgroundImage.Picture.LoadFro
SelectPalette(workdc,	mFile('androm.bmp');
backgroundImage.Picture.Bitmap.P	, , , , , , , , , , , , , , , , , , ,
alette, false);	paddle.Picture.LoadFromFile('paddl
workCanvas.Handle := workdc;	e.ico');
workCanvas.CopyRect(background	WindowState := wsMaximized;
Rect, backgroundCanvas,	,
backgroundRect);	backgroundRect.Top := 0;
workCanvas.Draw(0,0,	backgroundRect.Left := 0;
spriteImage.Picture.Icon);	backgroundRect.Right :=
paddleX := ClientWidth div 2;	ClientWidth;
paddleY := ClientHeight - 50;	backgroundRect.Bottom :=
workCanvas.Draw( paddleX,	ClientHeight;
paddleY, paddle.Picture.Icon);	9 /
paddleRect.Left := paddleX -	spriteRect.Top := 0;
paddle.Width;	spriteRect.Left := 0;
paddleRect.Right := paddleX +	spriteRect.Right :=
paddle.Width;	spriteImage.Picture.Width;
paddleRect.Top := paddleY;	spriteRect.Bottom :=
paddleRect.Bottom := paddleY +	spriteImage.Picture.Height;
paddle.Height;	
	//Set up backgroundCanvas
	backgrounddc :=

else	
Application.Terminate;	RealizePalette(backgroundCanvas.H
end;	andle);
enu,	andic),
/ > nwaaaduwa	RealizePalette(workCanvas.Handle);
<> procedure	Realizer arctic(workeanvas.frandic),
TForm1.IdleLoop(Sender: TObject;	Canvas.CopyRect(backgroundRect,
var Done: Boolean);	workCanvas, backgroundRect);
choice, SideDef, TopDef,	end;
1 - 1	cnu,
PaddleDifference: integer;	<> procedure
begin //Ivons loop going	TForm1.FormMouseMove(Sender:
//keeps loop going	TObject; Shift: TShiftState; X,
done := false;	Y: Integer);
//slows down action	begin
Sleep(1);	//Animates and moves paddle
ahangaDaata, aaataDaat	paddleCenter := X;
changeRect := spriteRect;	if(paddleCenter <
spriteRect.Left := x;	paddle.Picture.Width div 2) then
spriteRect.Top := y;	•
spriteRect.Right := x +	<pre>paddleCenter := paddle.Picture.Width div 2;</pre>
spriteImage.Picture.Width;	*
spriteRect.Bottom := y +	if(paddleCenter > ClientWidth -
spriteImage.Picture.Height;	(paddle.Picture.Width div 2)) then
	paddleCenter := ClientWidth -
workCanvas.CopyRect(paddleRect,	(paddle.Picture.Width div 2); end;
backgroundCanvas, paddleRect);	enu;
I DIUD ( IUD (	<> procedure
changePaddleRect := paddleRect;	TForm1.FormCreate(Sender:
paddleRect.Left := paddleCenter -	TObject);
((paddle.Picture.Width) div 2);	• / /
paddleX := paddleRect.Left;	//Assign idle time function
paddleRect.Top := paddleY;	Application.OnIdle := IdleLoop;
paddleRect.Right := paddleX +	Application.Ontule TuleLoop,
paddle.Picture.Width;	if(Application.MessageBox('Would
paddleRect.Bottom := paddleY +	you like to play with Earth?', 'Hello
paddle.Picture.Height;	_ · ·
	Earthling', MB_OKCANCEL) = IDOK) then
SideDef := changeRect.Left -	begin
spriteRect.Left;	//load sound effect
// If SideDiff < 0 the paddle is to the	sndPlaySound('Utopia
right	Default.wav', SND ASYNC or
if(SideDef < 0) then	SND FILENAME);
begin	
changeRect.Right :=	$\mathbf{x} := 0;$ $\mathbf{y} := 0.$
spriteRect.Right;	y := 0; EComoOvor := folso:
end	FGameOver := false;
else	ShowCursor(false);
begin	and
changeRect.Left :=	end

and ( (spriteRect.Right) >=
(paddleRect.Left))
and ( (spriteRect.Left) <=
(paddleRect.Right) ) then
begin
yDir := -5;
sndPlaySound('Utopia
Default.wav', SND_ASYNC or
SND FILENAME);
end;
,
if $(x \le 0)$ then
begin
xDir := 5;
,
end;
$if(x \ge ClientWidth - 16) then$
begin
xDir := -5;
end;
inc (x,xDir);
inc (y, yDir);
PaddleDifference :=
changePaddleRect.Left -
paddleRect.Left;
// If PaddleDiff < 0 the paddle is to
the right
8
if(PaddleDifference < 0) then
begin
changePaddleRect.Right :=
paddleRect.Right;
end
else
begin
changePaddleRect.Left :=
paddleRect.Left;
end;
,
//Perform dirty rectangle animation
on memory and Form canvas
workCanvas.Draw(x, y,
spriteImage.Picture.Icon);
workCanvas.Draw(paddleX,
paddleY, paddle.Picture.Icon);
RealizePalette(backgroundCanvas.H
andle);

spriteRect.Left;
end;
TopDef := changeRect.Top -
spriteRect.Top;
// If SideDiff < 0 the paddle is to the
Down
if(TopDef < 0) then
begin
changeRect.Bottom :=
spriteRect.Bottom;
end
else
begin
changeRect.Top :=
spriteRect.Top;
end;
workCanvas.CopyRect(spriteRect,
backgroundCanvas, spriteRect);
//ChangeRectCalcs
if (y <= 0) then
begin
yDir := 5;
end;
if (y >= ClientHeight - 16) then
begin
FGameOver := true;
SetCursor(HCURSOR(
IDC ARROW));
ShowCursor(true);
choice := MessageBox(Handle,
'You lost Earth', 'Try Again?',
MB_RETRYCANCEL);
if(choice = IDRETRY) then
begin
$\mathbf{x} := 0;$
y := 0;
ShowCursor(false);
end
else
Form1.Close;
end;
,
if ( (spriteRect.Bottom - 16) >=
(paddleRect.Top))
and ( (spriteRect.Bottom - 16) <=
(paddleRect.Top + 5))

OnActivate = FormActivate
OnCreate = FormCreate
OnMouseMove = FormMouseMove
OnPaint = FormPaint
PixelsPerInch = 96
TextHeight = 13
end

<> object <b>5</b>
الملف النصي لبرنامج
Football)
إعداد
علاء الدين اللباد
واتف ۱۷۳۵۷۵ <u>۶</u> ۰۹
<> object AboutForm:
TAboutForm
Left = 422
Top = 149
Width = 237
Height = 416
Caption = 'About Football'
Color = clBlue
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = False
Position = poScreenCenter
PixelsPerInch = 96
TextHeight = 13
<> object Label1: TLabel
Left = 29
Top = 6
Width = 171
Height = 16
Caption = 'FOOTBALL
INSTRUCTIONS'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWhite

RealizePalette(workCanvas.Handle);
Canvas.CopyRect(changeRect,
workCanvas, changeRect);
Canvas.CopyRect(changePaddleRec
t, workCanvas, changePaddleRect);
end;
<> procedure
TForm1.WMSetCursor(var
Message: TWMSetCursor);
begin
//Hides Cursor
if not(FGameOver) then
begin
SetCursor( HCURSOR( nil ) );
end;
end;
end.

<pre> &lt;&gt; object 1  EarthPng)  Alac  Alac  Palac  Pal</pre>	
EarthPng)  Alac]  Alac]  -9220V0FV1 - Alac]  -9220V0FV1 - Alac	<> object 1
الباد	الهلف النصي لبرناهج
علاء الدين اللباد  • • • • • • • • • • • • • • • • • • •	EarthPng)
-9220VOPV1 4:18  <> object Form1: TForm1  Left = 111  Top = 112  Width = 544  Height = 375  Caption = 'Earth Pong'  Color = clBtnFace  Font.Charset =  DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  Icon.Data = {  }	عداد
<pre>&lt;&gt; object Form1: TForm1 Left = 111 Top = 112 Width = 544 Height = 375 Caption = 'Earth Pong' Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Color = clWindowText Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] Icon.Data = { }</pre>	علاء الدين اللباد
Left = 111  Top = 112  Width = 544  Height = 375  Caption = 'Earth Pong'  Color = clBtnFace  Font.Charset =  DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  Icon.Data = {  }	الله ۱۷۳۵۷۵۶۹۰ هاتف ۱۷۳۵۷۵۶۹۰
Top = 112  Width = 544  Height = 375  Caption = 'Earth Pong'  Color = clBtnFace  Font.Charset =  DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  Icon.Data = {  }	<> object Form1: TForm1
Width = 544  Height = 375  Caption = 'Earth Pong'  Color = clBtnFace  Font.Charset =  DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  Icon.Data = {  }	Left = 111
Height = 375  Caption = 'Earth Pong'  Color = clBtnFace  Font.Charset =  DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  Icon.Data = {  }	Top = 112
Caption = 'Earth Pong' Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Color = clWindowText Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] Icon.Data = { }	Width = 544
Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Color = clWindowText Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] Icon.Data = { }	Height = 375
Font.Charset = DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  Icon.Data = { }	Caption = 'Earth Pong'
DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  Icon.Data = { }	Color = clBtnFace
Font.Color = clWindowText  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  Icon.Data = { }	Font.Charset =
Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] Icon.Data = { }	DEFAULT_CHARSET
Font.Name = 'MS Sans Serif' Font.Style = [] Icon.Data = { }	Font.Color = clWindowText
Font.Style = [] Icon.Data = { }	Font.Height = -11
Icon.Data = { }	Font.Name = 'MS Sans Serif'
}	Font.Style = []
} OldCreateOrder = False	Icon.Data = {
OldCreateOrder = False	}
	OldCreateOrder = False

•
one.'
'(Just kidding.)')
ParentCtl3D = False
ParentFont = False
ReadOnly = True
TabOrder = 0
end
<> object Button1: TButton
Left = 77
Top = 355
Width = 75
Height = 25
Caption = 'OK'
ModalResult = 1
TabOrder = 1
end
End
111111111111111111111111111111111111111

الهلف التنفيذي لبرنامج
40 procedures
Football)
إعداد
علاء الدين اللباد
واتف ۱۷۳۵۷۵ <u>۶</u> ۶۰
unit Main;
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs, Menus, ExtCtrls, StdCtrls,
ComCtrls, MMSystem;
cometris, minipystem,
const
RBTOP: integer = 58;
RBMIDDLE: integer = 74;
RBBOTTOM: integer = 90;
RBLEFT: integer = 36;
TACKLETOP: integer = 59;

Font.Height = -13
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
end
<> object Memo1: TMemo
Left = 18
Top = 27
Width = 192
Height = 321
BorderStyle = bsNone
Color = clBlue
Ctl3D = False
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWhite
Font.Height = -11
Font.Name = 'Default'
Font.Style = []
Lines.Strings = (
'The Computer''s on Defense.'
'You Control the Running Back.'
"
'Slide OFF switch to PRO 1'
'(PRO 2 for advanced players).'
" a n n n n n n n n n n n n n n n n n n
'Press ST button for downs, field'
'position and yards to go.'
'Press ARROW buttons (or
I,K,J,L'
'keys) to maneuver running back
'(bright blip) through tacklers '
'(dimmer blips). Tackled?'
'Game stops.'
"
'Press ST button to check status and'
'reset field. Try another play.'
11
'On fourth down, try a run, or
push'
'K button to punt or kick field
goal.'
11
'If the game malfunctions it may'
'mean battery wear. Use a fresh
Julion Julion

About1: TMenuItem;	<b>TACKLEMIDDLE</b> : integer = 7:
Clock: TTimer;	<b>TACKLEBOTTOM</b> : integer = 9
<> procedure	TACKLEROW1 : integer = 104;
About1Click(Sender: TObject);	TACKLEROW2 : integer = 148;
<> procedure	TACKLEROW3 : integer = 236;
Exit1Click(Sender: TObject);	
<> procedure	type
FormKeyPress(Sender: TObject;	spot = record
var Key: Char);	player : TLabel;
<> procedure	rb: boolean;
MoveUpClick(Sender: TObject);	end;
<> procedure	
MoveDownClick(Sender: TObject);	TMainForm = class(TForm)
<> procedure	Image7: TImage;
MoveForwardClick(Sender:	HomeDisplay: TLabel;
TObject);	TimeDisplay: TLabel;
<> procedure	VisitorDisplay: TLabel;
MoveBackClick(Sender: TObject);	Runningback: TLabel;
<> procedure	Tackler1: TLabel;
ComputerClick(Sender: TObject);	Tackler2: TLabel;
<> procedure	Tackler3: TLabel;
ClockTimer(Sender: TObject);	Tackler4: TLabel;
<> procedure	Tackler5: TLabel;
TimerTimer(Sender: TObject);	HomeLabel: TLabel;
<> procedure	TimeLabel: TLabel;
ScoreMouseUp(Sender: TObject;	VisitorLabel: TLabel;
Button: TMouseButton;	MoveForward: TImage;
Shift: TShiftState; X, Y:	MoveUp: TImage;
Integer);	MoveDown: TImage;
<> procedure	Kick: TImage;
ScoreMouseDown(Sender: TObject;	Score: TImage;
Button: TMouseButton; Shift: TShiftState; X, Y:	Label2: TLabel;
Integer);	Label1: TLabel;
<> procedure	Label3: TLabel;
KickClick(Sender: TObject);	DownDisplay: TLabel;
<> procedure	FieldPosDisplay: TLabel;
OnOffSwitchChange(Sender:	YTGDisplay: TLabel;
TObject);	DownLabel: TLabel;
private	FieldPosLabel: TLabel;
{ Private declarations }	YTGLabel: TLabel;
field: array[09] of array[02] of	MoveBack: TImage;
spot; // the X and Y coordinates of	OnOffSwitch: TTrackBar;
the field	Computer: TButton;
rbx, rby: integer; // the X and	Timer: TTimer;
Y coordinates of the runningback	MainMenu1: TMainMenu;
running: boolean;	File1: TMenuItem;
LastSack : TLabel;	Exit1: TMenuItem;
YardsToGo: integer;	Help1: TMenuItem;
1.114010001111005019	

'i':
MoveUpClick(Sender);
'k':
MoveDownClick(Sender);
'I':
MoveForwardClick(Sender);
'j':
MoveBackClick(Sender);
end;
end;
·
<> procedure
TMainForm.MoveUpClick(Sender:
TObject);
begin
if ((OnOffSwitch.Position = 2) or
(LastSack <> Nil)) then
exit;
if not running then
TogglePlay(true);
if (Runningback.Top = RBTOP)
then
exit; // already as up as it gets
if (field[rbx][rby-1].player <> Nil)
then
begin
Sacked(field[rbx][rby-1].player);
exit;
end;
if (Runningback.Top =
RBBOTTOM) then
Runningback.Top := RBMIDDLE
else if (Runningback.Top =
RBMIDDLE) then
Runningback.Top := RBTOP;
field[rbx][rby].player := Nil;
field[rbx][rby].rb := false;
dec( rby );
field[rbx][rby].player :=
Runningback;
field[rbx][rby].rb := true;
end;

Down: integer; FieldPosition: integer; Home: integer; Visitor: integer; Quarter: integer; TimeLeft: double; <> procedure ShowField(visible: boolean); <> procedure ResetField; <> procedure Sacked(Player: TLabel); <> procedure TogglePlay(toggle: boolean); <> procedure TogglePlay(toggle: boolean); <> procedure ResetGame; public { Public declarations } end;  var  MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	
Home: integer;  Visitor: integer;  Quarter: integer;  TimeLeft: double;  <> procedure ShowField(visible: boolean);  <> procedure ResetField;  <> procedure Sacked(Player: TLabel);  <> procedure TogglePlay(toggle: boolean);  <> procedure ResetGame; public  { Public declarations }  end;  var  MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	Down : integer;
Visitor: integer; Quarter: integer; TimeLeft: double; <> procedure ShowField(visible: boolean); <> procedure ResetField; <> procedure Sacked(Player: TLabel); <> procedure TogglePlay(toggle: boolean); <> procedure ResetGame; public { Public declarations } end;  var MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	FieldPosition: integer;
Quarter: integer; TimeLeft: double; <> procedure ShowField(visible: boolean); <> procedure ResetField; <> procedure Sacked(Player: TLabel); <> procedure ShowDisplay(visible: boolean); <> procedure TogglePlay(toggle: boolean); <> procedure ResetGame; public { Public declarations } end;  var MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	Home : integer;
Quarter: integer; TimeLeft: double; <> procedure ShowField(visible: boolean); <> procedure ResetField; <> procedure Sacked(Player: TLabel); <> procedure ShowDisplay(visible: boolean); <> procedure TogglePlay(toggle: boolean); <> procedure ResetGame; public { Public declarations } end;  var MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	
TimeLeft: double;  <> procedure ShowField(visible: boolean);  <> procedure ResetField;  <> procedure Sacked(Player : TLabel);  <> procedure ShowDisplay(visible: boolean);  <> procedure TogglePlay( toggle: boolean);  <> procedure ResetGame; public  { Public declarations } end;  var  MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	
ShowField(visible: boolean);  <> procedure ResetField;  <> procedure Sacked(Player: TLabel);  <> procedure ShowDisplay(visible: boolean);  <> procedure TogglePlay( toggle: boolean);  <> procedure ResetGame; public  { Public declarations } end;  var  MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	
ShowField(visible: boolean);  <> procedure ResetField;  <> procedure Sacked(Player: TLabel);  <> procedure ShowDisplay(visible: boolean);  <> procedure TogglePlay( toggle: boolean);  <> procedure ResetGame; public  { Public declarations } end;  var  MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	,
<> procedure ResetField; <> procedure Sacked(Player : TLabel); <> procedure ShowDisplay(visible: boolean); <> procedure TogglePlay( toggle: boolean); <> procedure ResetGame; public { Public declarations } end;  var  MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end; <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end; <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	
<pre>&lt;&gt; procedure Sacked(Player : TLabel);     &lt;&gt; procedure ShowDisplay(visible : boolean);     &lt;&gt; procedure TogglePlay( toggle : boolean );     &lt;&gt; procedure ResetGame; public     { Public declarations }     end;  var     MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  &lt;&gt; procedure TMainForm.About1Click(Sender: TObject); begin     AboutForm.ShowModal; end;  &lt;&gt; procedure TMainForm.Exit1Click(Sender: TObject); begin     Application.Terminate; end;  &lt;&gt; procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin</pre>	
: TLabel);  <> procedure ShowDisplay(visible: boolean);  <> procedure TogglePlay( toggle: boolean);  <> procedure ResetGame; public  { Public declarations } end;  var  MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	
<pre>&lt;&gt; procedure ShowDisplay(visible : boolean);     &lt;&gt; procedure TogglePlay(     toggle : boolean );     &lt;&gt; procedure ResetGame;     public         { Public declarations }         end;  var         MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  &lt;&gt; procedure         TMainForm.About1Click(Sender:         TObject);         begin         AboutForm.ShowModal;         end;  &lt;&gt; procedure         TMainForm.Exit1Click(Sender:         TObject);         begin         Application.Terminate;         end;  &lt;&gt; procedure         TMainForm.FormKeyPress(Sender:         TObject; var Key: Char);         begin</pre>	_ ` <u> </u>
ShowDisplay(visible: boolean);  <> procedure TogglePlay( toggle: boolean);  <> procedure ResetGame; public  { Public declarations } end;  var  MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	
<> procedure TogglePlay( toggle: boolean); <> procedure ResetGame; public { Public declarations } end;  var  MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	l =
toggle: boolean);  <> procedure ResetGame; public { Public declarations } end;  var  MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	
<pre></pre>	
public { Public declarations } end;  var  MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	
{ Public declarations } end;  var  MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  <> procedure  TMainForm.About1Click(Sender:  TObject); begin  AboutForm.ShowModal; end;  <> procedure  TMainForm.Exit1Click(Sender:  TObject); begin  Application.Terminate; end;  <> procedure  TMainForm.FormKeyPress(Sender:  TObject; var Key: Char); begin	
end;  var  MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  <> procedure  TMainForm.About1Click(Sender:  TObject);  begin    AboutForm.ShowModal; end;  <> procedure  TMainForm.Exit1Click(Sender:  TObject);  begin    Application.Terminate; end;  <> procedure  TMainForm.FormKeyPress(Sender:  TObject; var Key: Char); begin	-
var  MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	
MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	_ chu,
MainForm: TMainForm;  implementation  uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	var
implementation  uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	
uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	Traini or in,
uses about;  {\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	implementation
{\$R *.dfm}  <> procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	Implementation
{\$R *.dfm}  <> procedure   TMainForm.About1Click(Sender:   TObject);   begin   AboutForm.ShowModal;   end;  <> procedure   TMainForm.Exit1Click(Sender:   TObject);   begin   Application.Terminate;   end;  <> procedure   TMainForm.FormKeyPress(Sender:   TObject; var Key: Char);   begin	usos about
<pre>&lt;&gt; procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end; &lt;&gt; procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end; &lt;&gt; procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin</pre>	uses about,
<pre>&lt;&gt; procedure TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end; &lt;&gt; procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end; &lt;&gt; procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin</pre>	(\$D * dfm)
TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	(JK .uim)
TMainForm.About1Click(Sender: TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	< > mmaaaduma
TObject); begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	<u> </u>
begin AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	
AboutForm.ShowModal; end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	
end;  <> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	
<> procedure TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end; <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	,
TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	end;
TMainForm.Exit1Click(Sender: TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	•
TObject); begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	l -
begin Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	`
Application.Terminate; end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	
end;  <> procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	
<pre>&lt;&gt; procedure TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin</pre>	·
TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	end;
TMainForm.FormKeyPress(Sender: TObject; var Key: Char); begin	
TObject; var Key: Char); begin	<u> </u>
begin	· ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` `
TZ P	
case Key of	case Key of

TogglePlay(true);
if (Runningback,Left <
TACKLEROW3 - 2) then
begin
if (field[rbx+1][rby].player <> Nil)
then
begin
Sacked(field[rbx+1][rby].player);
exit;
end;
field[rbx][rby].player := Nil;
field[rbx][rby].rb := false;
Runningback.Left :=
Runningback.Left + 22;
inc(rbx);
end
else
begin
if (field[0][rby].player <> Nil) then
begin
Sacked(field[0][rby].player);
exit; end;
field[rbx][rby].player := Nil;
field[rbx][rby].rb := false;
Runningback.Left := RBLEFT;
rbx := 0;
end;
dec( YardsToGo );
inc( FieldPosition );
if (FieldPosition = 100) then
begin
sndPlaySound('td.wav',
SND_SYNC);
running := false;
ShowField(false);
inc( Home, 7);
HomeDisplay.Caption :=
inttostr(Home);
FieldPosition:=80; // Starting
position for visitor
Down := 1;
YardsToGo := 10;
Computer.Visible := true;
exit;
end;

<> procedure
TMainForm.MoveDownClick(Sende
r: TObject);
begin
if ((OnOffSwitch.Position = 2) or
(LastSack <> Nil)) then
exit;
if not running then
TogglePlay(true);
roggici lay(truc);
if (Dunninghealt Ton -
if (Runningback.Top =
RBBOTTOM) then
exit; // already as down as it gets
10/00 111 111 141 141
if (field[rbx][rby+1].player <> Nil)
then
begin
Sacked(field[rbx][rby+1].player);
exit;
end;
if (Runningback.Top = RBTOP)
then
Runningback.Top :=
RBMIDDLE
else if (Runningback.Top =
RBMIDDLE) then
Runningback.Top :=
RBBOTTOM;
field[rbx][rby].player := Nil;
field[rbx][rby].rb := false;
inc(rby);
field[rbx][rby].player :=
Runningback;
field[rbx][rby].rb := true;
end;
cnu,
<> procedure
TMainForm.MoveForwardClick(Se
nder: TObject);
begin
if ((OnOffSwitch.Position = 2) or
(LastSack <> Nil)) then
exit;
if not running then

Inc( YardsToGo );
Dec( FieldPosition );
Dec( Field) osition ),
Cold(why)(why) whosen
field[rbx][rby].player :=
Runningback;
field[rbx][rby].rb := true;
end;
end;
<> procedure
TMainForm.ResetField;
var
x, y :integer;
begin
TogglePlay(false);
LastSack := Nil;
// empty the field
for y := 0 to 2 do
for x := 0 to 9 do
begin
field[x][y].player := Nil;
field[x][y].rb := false;
end;
//:-:4:-114:
// initial locations of players
field[0][1].player := Runningback;
field[0][1].rb := true;
rbx := 0;
rby := 1;
field[3][0].player := Tackler1;
field[3][1].player := Tackler2;
field[3][2].player := Tackler3;
field[5][1].player := Tackler4;
field[9][1].player := Tackler5;
,
Runningback.Left := RBLEFT;
Runningback.Top := RBMIDDLE;
Tackler1.Left := TACKLEROW1;
Tackler1.Top := TACKLEROW1,  Tackler1.Top := TACKLETOP;
Tackler2.Left := TACKLEROW1;
Tackler2.Top :=
TACKLEMIDDLE;
Tackler3.Left := TACKLEROW1;
Tackler3.Top :=
TACKLEBOTTOM;
Tackler4.Left := TACKLEROW2;
Tackler4.Top :=

field[rbx][rby].player :=
Runningback;
field[rbx][rby].rb := true;
end;
- Chu,
<> procedure
_
TMainForm.MoveBackClick(Sender
: TObject);
begin
if ((OnOffSwitch.Position = 2) or
(LastSack <> Nil)) then
exit;
if not running then
TogglePlay(true);
if(FieldPosition > 0) then
begin
if (Runningback.Left > RBLEFT
+ 2) then
,
begin
if (field[rbx-1][rby].player <>
Nil) then
begin
Sacked(field[rbx-
1][rby].player);
exit;
end;
field[rbx][rby].player := Nil;
field[rbx][rby].rb := false;
Runningback.Left :=
Runningback.Left - 22;
dec( rbx );
end
else
begin
if (field[9][rby].player <> Nil)
then
begin
Sacked(field[9][rby].player);
exit;
end;
field[rbx][rby].player := Nil;
field[rbx][rby].rb := false;
Runningback.Left :=
TACKLEROW3;
rbx := 9;
end;

TA CIVI EMIDDI E
TACKLEMIDDLE;
Tackler5.Left := TACKLEROW3;
Tackler5.Top :=
TACKLEMIDDLE;
,
ShowField(true);
` /
end;
<> procedure
TMainForm.Sacked( Player:
TLabel);
begin
sndPlaySound('whistle.wav',
SND SYNC);
running := false;
LastSack := player;
if (YardsToGo <=0) then
begin
Down := 1;
YardsToGo := 10;
end
else
begin
inc( Down );
if (Down > 4) then
begin
sndPlaySound('whistle.wav',
SND SYNC);
Down := 1; // First down for
,
visitor
YardsToGo := 10;
Computer.Visible := true;
end;
end;
end;
· · · · · · · · · · · · · · · · · · ·
<> procedure
TMainForm.ShowDisplay(visible:
boolean);
begin
DownLabel.Visible := visible;
FieldPosLabel.Visible := visible;
YTGLabel.Visible := visible;
DownDisplay.Visible := visible;
FieldPosDisplay.Visible := visible;
YTGDisplay.Visible := visible;
end;

if (LastSack <> Nil) then
LastSack.Visible := not
LastSack.Visible;
exit;
end;
if (field[x][y].rb) then
exit; // can't move the
runningback!
if (field[x][y].player = Nil) then
exit; // no tacker at this spot
•
if (field[x][y].player.Left <=
RBLEFT) then
exit; // already at the endzone
if (direction = 0) then// we're
moving horizontal
begin
if (x < rbx) then
newx := x + 1
else if $(x > rbx)$ then
newx := x - 1
else
exit;
// we're already horizontally lined
up with rb
end
else if (direction = 1) then// we're
moving vertical
begin
if (y < rby) then
newy := y + 1
else if $(y > rby)$ then
newy := y - 1
else
exit;
// we're already vertically lined
up with rb
end;
if field[newx][newy].rb then// got
him!
begin
Sacked(field[x][y].player);
exit;
end;
ciiu,

TMainForm.ClockTimer(Sender:
TObject);
begin
if not running then
exit;
sndPlaySound('tick.wav',
SND_ASYNC);
TimeLeft := TimeLeft - 0.1;
TimeDisplay.Caption :=
FloatToStrF(TimeLeft, ffGeneral, 4,
4);
if (TimeLeft <= 0) then
begin
inc( Quarter );
TimeLeft := 15;
end;
if (Quarter >= 5) then
begin
// game over
sndPlaySound('whistle.wav',
SND_SYNC);
sndPlaySound('whistle.wav',
SND_SYNC);
LastSack := Runningback; // hack
to keep movement keys disabled
TogglePlay(false);
ShowField(false);
ShowDisplay(true);
end;
end;
<> procedure
TMainForm.TimerTimer(Sender:
TObject);
var
x, y, newx, newy, direction :
integer;
begin
newy := random(3);
y := newy;
newx := random(10);
x := newx;
direction := random(2); $// 0$ is for x,
1 is for y
if not running then
begin

IntToStr(FieldPosition) + ' <';	
end	
else	
begin	
FieldPosDisplay.Caption := '> ' +	
IntToStr(100 - FieldPosition);	
end;	
DownDisplay.Caption :=	
IntToStr(Down);	
ShowField(false);	
ShowDisplay(true);	
end;	
Ź	
<> procedure	
TMainForm.KickClick(Sender:	
TObject);	
begin	
if ((running) or (Down <> 4)) then	
exit; // button only valid just	
before fourth down	
ShowField(false);	
inc(FieldPosition,random(100));	
if (FieldPosition >= 100) then	
begin	
sndPlaySound('td.wav',	
SND SYNC);	
inc( Home, 3 );	
HomeDisplay.Caption :=	
IntToStr(Home);	
` ''	
FieldPosition := 80; // Starting	
position for visitor	
end	
else	
begin IN C 1/1 1: 4	
sndPlaySound('whistle.wav',	
SND_SYNC);	
sndPlaySound('whistle.wav',	
SND_SYNC);	
end;	
Down := 1; // first down for visitor	
YardsToGo := 10;	
Computer.Visible := true;	
LastSack := Runningback; // hack	
to keep movement keys disabled	
end;	

if (field[newx][newy].player = Nil)
then// not blocked
begin
field[x][y].player.Left :=
field[x][y].player.Left - (22 * (x -
newx));
field[x][y].player.Top :=
field[x][y].player.Top - (16 * (y -
newy));
field[newx][newy].player :=
field[x][y].player;
field[x][y].player := Nil;
end;
end;
< > procedure
<pre>&lt;&gt; procedure TMain Form Soon Mongo Un(Sondon)</pre>
TMainForm.ScoreMouseUp(Sender:
TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin
if ((OnOffSwitch.Position = 2) or
(running)) then
exit;
CAIL
ChowDianlay(falsa)
ShowDisplay(false);
if not(Computer.Visible) then
ResetField;
end;
<> procedure
TMainForm.ScoreMouseDown(Send
er: TObject; Button:
TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin
if ((OnOffSwitch.Position = 2) or
(running)) then
exit;
if (Quarter = 5) then
exit;
CAIL,
VTCD:sular Couting
YTGDisplay.Caption :=
IntToStr(YardsToGo);
if (FieldPosition <= 50) then
begin
FieldPosDisplay.Caption :=
riciar ospispiay. Caption .

randomize;
ResetField;
end;
enu,
end.
enu.
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
unit about;
unit about;
interface
uses
Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
StdCtrls;
stuctris,
type
TAboutForm = class(TForm)
Label1: TLabel;
Memo1: TMemo;
Button1: TButton;
private
{ Private declarations }
public
{ Public declarations }
end;
/
var
AboutForm: TAboutForm;
1100001 011110 11100 011111
implementation
{\$R *.dfm}
•
end.
\\\\\\\\\\

<> procedure
TMainForm.OnOffSwitchChange(S
ender: TObject);
begin
case OnOffSwitch.Position of
1:
begin
Timer.Interval := 250;
HomeDisplay.Visible := true;
VisitorDisplay.Visible := true;
TimeDisplay.Visible := true;
ResetGame;
end;
2:
begin
ShowField(false);
HomeDisplay.Visible := false;
VisitorDisplay. Visible := false;
TimeDisplay. Visible := false;
end;
3:
begin Times Laterand 100
Timer.Interval := 100;
HomeDisplay.Visible := true;
VisitorDisplay.Visible := true;
TimeDisplay.Visible := true;
ResetGame;
end;
end;
end;
<> procedure
TMainForm.ResetGame;
begin
YardsToGo := 10;
Down := 1;
FieldPosition := 20;
Home := 0;
Visitor := 0;
Quarter := 1;
TimeLeft := 15;
TimeDisplay.Caption :=
FloatToStrF(TimeLeft, ffGeneral, 4,
4);
HomeDisplay.Caption :=
IntToStr(Home);
VisitorDisplay.Caption :=
intToStr(Visitor);

# القسم السابع مجموعة من البرامج المتقدمة والشاملة في دلفي (متقدمين)

<b>ObjectS</b>
الهلف التنفيذي لبرناهج
(AlphaBlendDemo)
اعداد عامدار
علاء الدين اللباد
واتف ۷۳۵۷۵۶۹۰
AlphaBlendDemo
49
program AlphaBlendDemo;

TRichEditUnderline;	uses
RichEditStrikeOut1:	Forms,
TRichEditStrikeOut;	Main i
RichEditBullets1:	
TRichEditBullets;	{\$R *.re
RichEditAlignLeft1:	
TRichEditAlignLeft;	begin
RichEditAlignRight1:	Applica
TRichEditAlignRight;	Applica
RichEditAlignCenter1:	Form1);
TRichEditAlignCenter;	Applic
FileExit1: TFileExit;	end.
SearchFind1: TSearchFind; SearchFindNext1:	
TSearchFindNext;	*************
· · · · · · · · · · · · · · · · · · ·	\\\\\\\
SearchReplace1: TSearchReplace; SearchFindFirst1:	_ • 4 Th #F
TSearchFindFirst;	unit Ma
ActionToolBar1:	: 4 <b>C</b>
TActionToolBar;	interfac
procedure FormCreate(Sender:	Nege
TObject);	uses Windo
private	Variant
{ Private declarations }	Control
procedure PopupClass(Sender:	Dialogs
TObject; var PopupClass:	ActnLis
TCustomPopupClass);	ActnCtr
public	ActnM
{ Public declarations }	ImgList
end;	XPStyl
{ TCustomLayeredActionPopupEx }	type
	TForm
TCustomLayeredActionPopupEx =	Action
class(TXPStylePopupMenu)	<b>TAction</b>
private	Action
FAlphaBlend: Boolean;	<b>TAction</b>
FAlphaBlendValue: Byte;	RichE
FTransparentColor: Boolean;	Image
FTransparentColorValue:	EditC
TColor;	EditC
procedure SetAlphaBlend(const	EditP
Value: Boolean);	EditS
procedure	EditU
SetAlphaBlendValue(const Value: Byte);	EditD
procedure	RichE
SetTransparentColor(const Value:	RichE
Boolean);	RichE
Doucaii),	

uses
Forms,
Main in 'Main.pas' {Form1};
main in main.pas (Formi),
{\$R *.res}
begin
Č
Application.Initialize;
Application.CreateForm(TForm1,
Form1);
Application.Run;
end.
enu.
\\\\\\
weit Main
unit Main;
interface
uses
Windows, Messages, SysUtils,
Variants, Classes, Graphics,
Controls, Forms,
Dialogs, StdActns, ExtActns,
ActnList, ToolWin, ActnMan,
ActnCtrls,
ActnMenus, StdCtrls, ComCtrls,
ImgList, StdActnMenus,
XPStyleActnCtrls, XPActnCtrls;
,
two
type
TForm1 = class(TForm)
ActionManager1:
TActionManager;
ActionMainMenuBar1:
TActionMainMenuBar;
,
RichEdit1: TRichEdit;
ImageList1: TImageList;
EditCut1: TEditCut;
EditCopy1: TEditCopy;
EditPaste1: TEditPaste;
EditSelectAll1: TEditSelectAll;
EditUndo1: TEditUndo;
EditDelete1: TEditDelete;
RichEditBold1: TRichEditBold;
RichEditItalic1: TRichEditItalic;
RichEditUnderline1:

	procedure
procedure	SetTransparentColorValue(const
TCustomLayeredActionPopupEx.Se	Value: TColor);
tAlphaBlendValue(const Value:	protected
Byte);	procedure CreateParams(var
begin	Params: TCreateParams); override;
if FAlphaBlendValue  Value	procedure
then	CreateWindowHandle(const
begin	Params: TCreateParams); override;
FAlphaBlendValue := Value;	procedure InitAlphaBlending(var
SetLayeredAttribs;	Params: TCreateParams);
end;	procedure SetLayeredAttribs;
end;	property AlphaBlend: Boolean
enu,	read FAlphaBlend write
nwaaaduwa	SetAlphaBlend;
procedure TCustomLayeredActionPopupEx.Se	property AlphaBlendValue: Byte
tLayeredAttribs;	read FAlphaBlendValue write
*	SetAlphaBlendValue;
const cUseAlpha: array [Boolean] of	property TransparentColor:
Integer = (0, LWA_ALPHA);	Boolean read FTransparentColor
cUseColorKey: array [Boolean] of	write SetTransparentColor;
Integer = (0, LWA_COLORKEY);	property TransparentColorValue:
	TColor read
A Styles Integers	FTransparentColorValue write
AStyle: Integer;	SetTransparentColorValue;
begin	public
if not (csDesigning in	constructor Create(AOwner:
ComponentState) and	TComponent); override;
(Assigned (Set I expend) Window Attrib	end;
(Assigned(SetLayeredWindowAttrib utes)) and HandleAllocated then	
- //	var
begin AStyle :=	Form1: TForm1;
GetWindowLong(Handle,	101111111111111111111111111111111111111
GWL EXSTYLE);	implementation
if FAlphaBlend then	Implementation
begin	{\$R *.dfm}
if (AStyle and	(the sum)
WS EX LAYERED) = 0 then	{ TCustomLayeredActionPopup }
SetWindowLong(Handle,	(1 Custominayer currenom opup }
GWL EXSTYLE, AStyle or	procedure
WS EX LAYERED);	TCustomLayeredActionPopupEx.Se
WS_EA_LATERED),	tAlphaBlend(const Value: Boolean);
SetLayeredWindowAttributes(Hand	begin
le, FTransparentColorValue,	if FAlphaBlend <> Value then
FAlphaBlendValue,	
cUseAlpha[FAlphaBlend] or	begin  EAlphaPland: Value:
cUseColorKey[FTransparentColor])	FAlphaBlend := Value; SetLayeredAttribs;
1 1	•
end	end;
CHU	end;

utes)) then	else
if FAlphaBlend or	
FTransparentColor then	begin
	SetWindowLong(Handle,
Params.ExStyle :=	GWL_EXSTYLE, AStyle and not
Params.ExStyle or	WS_EX_LAYERED);
WS_EX_LAYERED;	RedrawWindow(Handle, nil, 0,
end;	RDW_ERASE or
	RDW_INVALIDATE or
procedure	RDW_FRAME or
TCustomLayeredActionPopupEx.Cr	RDW_ALLCHILDREN);
eateWindowHandle(	end;
const Params: TCreateParams);	end;
begin	end;
inherited;	
SetLayeredAttribs;	procedure
end;	TCustomLayeredActionPopupEx.Se
,	tTransparentColorValue(const
constructor	Value: TColor);
TCustomLayeredActionPopupEx.Cr	begin
eate(AOwner: TComponent);	if FTransparentColorValue <>
begin	Value then
inherited;	begin
,	FTransparentColorValue :=
FAlphaBlend := True;	Value;
FAlphaBlendValue := 125;	,
FTransparentColorValue :=	SetLayeredAttribs;
clWhite;	end;
end;	end;
procedure	procedure
TCustomLayeredActionPopupEx.Cr	TCustomLayeredActionPopupEx.Se
eateParams(	tTransparentColor(const Value:
var Params: TCreateParams);	Boolean);
begin	begin
inherited;	if FTransparentColor <> Value
InitAlphaBlending(Params);	then
end;	begin
	FTransparentColor := Value;
procedure	SetLayeredAttribs;
TForm1.FormCreate(Sender:	end;
TObject);	end;
begin	,
~~8^^	procedure
ActionMainMenuBar1.OnGetPopup	TCustomLayeredActionPopupEx.Ini
Class := PopupClass;	tAlphaBlending(var Params:
	TCreateParams);
end;	begin
www.aaduus	if not (csDesigning in
procedure	ComponentState) and
TForm1.PopupClass(Sender:	
TObject;	(assigned(SetLayeredWindowAttrib

end.	
111111111111111111111111111111111111111	

ColorMap.HighlightColor =
15660791
ColorMap.BtnSelectedColor =
clBtnFace
ColorMap.UnusedColor =
15660791
EdgeBorders = [ebTop, ebBottom]
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'Tahoma'
Font.Style = []
Spacing = 3
end
object RichEdit1: TRichEdit
Left = 0
Top = 53
Width = 413
Height = 234
Align = alClient
Lines.Strings = (
'RichEdit1')
TabOrder = 1
end
object ActionToolBar1:
TActionToolBar
Left = 0
Top = 27
Width = 413
Height = 26
ActionManager =
ActionManager1
Caption = 'ActionToolBar1'
ColorMap.HighlightColor =
15660791
ColorMap.BtnSelectedColor =
clBtnFace
ColorMap.UnusedColor =
15660791
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText

,	var PopupClass:
	CustomPopupClass);
b	egin
	PopupClass :=
T	CustomLayeredActionPopupEx;
e	nd;

ObjectS
الهلف االنصي لبرناهج
(AlphaBlendDemo)
إعداد
علاء الدين اللباد
واتة. ۷۳°400336٠
AlphaBlendDemo
object Form1: TForm1
Left = 241
Top = 161
Width = 421
Height = 321
Caption = 'ActionBand
AlphaBlended Menus Demo'
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = False
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
object ActionMainMenuBar1:
TActionMainMenuBar
Left = 0
Top = 0
Width = 413
Height = 27
UseSystemFont = False
ActionManager =
ActionManager1
AnimationStyle = asNone
Caption = 'ActionMainMenuBar1'

ShortCut = 46
end>
Caption = '&Edit'
end
item
Items = <
item
Action = SearchFind1
ImageIndex = 14
ShortCut = 16454
end
item
Action = SearchFindNext1
ImageIndex = 15
ShortCut = 114
end
item
Action = SearchReplace1
ImageIndex = 16
end
item
Action = SearchFindFirst1
end>
Caption = '&Search'
end
item
Items = <
item
Action = RichEditBold1
ImageIndex = 5
ShortCut = 16450
end
item
Action = RichEditItalic1
ImageIndex = 6
ShortCut = 16457
end
item
Action =
RichEditUnderline1
ImageIndex = 7
ShortCut = 16469
end
item
Action =
RichEditStrikeOut1
ImageIndex = 8
end

Ford Height = 11
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
Spacing = 0
end
object ActionManager1:
TActionManager
ActionBars = <
item
Items = <
item
Items = <
item
Action = FileExit1
ImageIndex = 13
end>
Caption = '&File'
end
item
Items = <
item
Action = EditCut1
ImageIndex = 0
<b>ShortCut = 16472</b>
end
item
Action = EditCopy1
ImageIndex = 1
ShortCut = 16451
end
item
Action = EditPaste1
ImageIndex = 2
<b>ShortCut = 16470</b>
end
item
Action = EditSelectAll1
<b>ShortCut = 16449</b>
end
item
Action = EditUndo1
ImageIndex = 3
ShortCut = 16474
end
item
Action = EditDelete1
ImageIndex = 4

item
Action = RichEditBold1
Caption = 'B&old'
ImageIndex = 5
ShortCut = 16450
end>
ActionBar = ActionToolBar1
end>
Images = ImageList1
Left = 135
Top = 164
StyleName = 'XP Style'
object EditCut1: TEditCut
Category = 'Edit'
Caption = 'Cu&t'
Hint = 'Cut Cuts the selection
and puts it on the Clipboard'
ImageIndex = 0
<b>ShortCut = 16472</b>
end
object EditCopy1: TEditCopy
Category = 'Edit'
Caption = '&Copy'
Hint = 'Copy Copies the selection
and puts it on the Clipboard'
ImageIndex = 1
ShortCut = 16451
end
object EditPaste1: TEditPaste
Category = 'Edit'
Caption = '&Paste'
Hint = 'Paste Inserts Clipboard
contents'
ImageIndex = 2
ShortCut = 16470
end
object EditSelectAll1:
TEditSelectAll
Category = 'Edit'
Caption = 'Select &All'
Hint = 'Select All Selects the
entire document'
ShortCut = 16449
end
object EditUndo1: TEditUndo
Category = 'Edit'
Caption = '&Undo'
Hint = 'Undo Reverts the last

_
item
Action = RichEditBullets1
Caption = 'Bu&llets'
ImageIndex = 9
end
item
Action =
RichEditAlignLeft1
Caption = '&Align Left'
ImageIndex = 10
end
item
Action =
RichEditAlignRight1
ImageIndex = 11
end
item
Action =
RichEditAlignCenter1
ImageIndex = 12
end>
Caption = 'F&ormat'
end>
ActionBar =
ActionMainMenuBar1
end
item
Items = <
item
Action =
1
RichEditAlignCenter1
ImageIndex = 12
end
item
Action =
RichEditAlignRight1
ImageIndex = 11
end
item
Action = RichEditUnderline1
ImageIndex = 7
ShortCut = 16469
end
item
Action = RichEditItalic1 ImageIndex = 6
i mageingev = 6
ShortCut = 16457

TRichEditBullets
Category = 'Format'
AutoCheck = True
Caption = '&Bullets'
Hint = 'Bullets Inserts a bullet on
the current line'
ImageIndex = 9
end
object RichEditAlignLeft1:
TRichEditAlignLeft
Category = 'Format' AutoCheck = True
AutoCheck = True
Caption = 'Align &Left'
Hint = 'Align Left Aligns text at
the left indent'
ImageIndex = 10
end
object RichEditAlignRight1:
TRichEditAlignRight
Category = 'Format'
AutoCheck = True
Caption = 'Align & Right'
Hint = 'Align Right Aligns text at
the right indent'
ImageIndex = 11
end
object RichEditAlignCenter1:
· ·
TRichEditAlignCenter
Category = 'Format'
AutoCheck = True
Caption = '&Center'
Hint = 'Center Centers text
between margins'
ImageIndex = 12
end
object FileExit1: TFileExit
Category = 'File'
Caption = 'E&xit'
Hint = 'Exit Quits the
application'
ImageIndex = 13
end
object SearchFind1: TSearchFind
Category = 'Search'
Caption = '&Find'
Hint = 'Find Finds the specified
text'
ImageIndex = 14
imagemuex – 14

action'
ImageIndex = 3
<b>ShortCut</b> = 16474
end
object EditDelete1: TEditDelete
Category = 'Edit'
Caption = '&Delete'
Hint = 'Delete Erases the
selection'
ImageIndex = 4
ShortCut = 46
end
object RichEditBold1:
TRichEditBold
Category = 'Format'
AutoCheck = True
Caption = '&Bold'
Hint = 'Bold'
ImageIndex = 5
ShortCut = 16450
end
object RichEditItalic1:
TRichEditItalic
Category = 'Format'
AutoCheck = True
Caption = '&Italic'
Hint = 'Italic'
ImageIndex = 6
ShortCut = 16457
end
object RichEditUnderline1:
TRichEditUnderline
Category = 'Format'
AutoCheck = True
Caption = '&Underline'
Hint = 'Underline'
ImageIndex = 7
ShortCut = 16469
end
object RichEditStrikeOut1:
TRichEditStrikeOut
Category = 'Format'
AutoCheck = True
Caption = '&Strikeout'
Hint = 'Strikeout'
ImageIndex = 8
end
object RichEditBullets1:

uses
Forms,
main in 'main.pas' {Form1};
{\$R *.res}
hagin
begin Application.Initialize;
Application.TreateForm(TForm1,
Form1);
Application.Run;
end.
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
unit main;
interface
uses
Windows, Messages, SysUtils,
Variants, Classes, Graphics,
Controls, Forms,
Dialogs, ActnMan, ActnList,
ToolWin, ActnCtrls, ActnMenus, StdCtrls,
StdActns, ExtCtrls, ComCtrls,
XPStyleActnCtrls;
Ai StyleActifettis,
type
TForm1 = class(TForm)
ActionManager1:
TActionManager;
Action1: TAction;
Action2: TAction;
ActionMainMenuBar1:
TActionMainMenuBar;
EditCut1: TEditCut;
EditCopy1: TEditCopy;
EditPaste1: TEditPaste;
EditSelectAll1: TEditSelectAll;
EditUndo1: TEditUndo;
Memo1: TMemo;
Panel1: TPanel;
Button4: TButton;
Button1: TButton;
Button3: TButton; Button2: TButton;
AddCategoryAction: TAction;
Audeauguryacuun, Tacuun,

ShortCut = 16454
end
object SearchFindNext1:
TSearchFindNext
Category = 'Search'
Caption = 'Find &Next'
Hint = 'Find Next Repeats the
last find'
ImageIndex = 15
ShortCut = 114
end
object SearchReplace1:
TSearchReplace
Category = 'Search'
Caption = '&Replace'
Hint = 'Replace Replaces specific
text with different text'
ImageIndex = 16
end
object SearchFindFirst1:
TSearchFindFirst
Category = 'Search'
Caption = 'F&ind First'
Hint = 'Find First Finds the first
occurance of specified text'
end
end
object ImageList1: TImageList
Left = 114
Top = 76
Bitmap = {
}
end
end

O	bjectS
ě	الهلف التنفيذي لبرناه
(]	DynaActionBands)
4	اعدإ
د	علاء الدين اللبا
٠	هاتف ۹٤٤٥٧٥٣٧١
Dy	naActionBands
pr	ogram DynaActionBands;

Item:= ActionManager1.FindItemByAction( ActionManager1.AddSeparator(Ite m); if Assigned(Item) then  ActionManager1.AddSeparator(Ite m); end; end; procedure TForm1.AddCategoryActionExecute (Sender: TObject); var Item: TActionClientItem; begin ActionManager1.FindItemByCaption n(*&text'); if Assigned(Item) then  ActionManager1.AddCategory('Edit ', Item); end; procedure TForm1.AddActionActionExecute(Sender: TObject); var Item: TActionClientItem; begin Item: ActionManager1.FindItemByCaption n(*&ActionManager1.FindItemByCaption n(*ActionManager1.FindItemByCaption n(*ActionManager1.FindItemSycaption n(*ActionManager1.FindItemSycaption n(*ActionManager1.FindItemSycaption n(*ActionManager1.FindItemSycaption n(*ActionItem: procedure TForm1.DeleteItemsActionExecute( Sender: TObject); begin  AddActionAction: TAction: ActionList; EditDelete2: TEditDelete procedure TObject); procedure  { Private declarations } public { Public declarations } procedure TForm1.DeleteItemsActionExecute( Sender: TObject); begin  ActionManager1.FindItemSycapt		
Action1);  if Assigned(Item) then  ActionManager1.AddSeparator(Item); end; end;  Item: TActionClientItem; begin Item: ActionManager1.FindItemByCaption ('&test'); end;  procedure TForm1.AddCategory('Edit', Item); end;  procedure TForm1.AddCationActionExecute(Sender: TObject); yar  Item: TActionClientItem; begin  Item: TActionClientItem;  Item: TActionClientItem;  Item: TActionClientItem;  Item: TActionClientItem;		,
if Assigned(Item) then  ActionManager1.AddSeparator(Item); end;  procedure  procedure  TForm1.AddCategoryActionExecute (Sender: TObject); var  Item: TActionClientItem; begin  Item: =  ActionManager1.FindItemByCaption n'(&test'); if Assigned(Item) then  ActionManager1.AddCategory('Edit', Item); end;  procedure  TForm1.AddActionActionExecute(Sender: TObject); var  Item: TActionClientItem; begin  procedure  ActionManager1.FindItemByCaption n'('&ActionI'); if Assigned(Item) then  ActionManager1.AddAction(EditCutl, Item); end;  procedure  TForm1.DisableBtnClick(Sender: TObject); begin  ActionManager1.FindItemByCaption n'('Editon'); if Assigned(Item) then  ActionManager1.FindItemByCaption n'('Editon'); if Assigned(Item) then  ActionManager1.AddAction(EditCutl, Item); end;  procedure  TForm1.DisableBtnClick(Sender: Tobject); begin  ActionManager1.DeleteActionItems(ActionI); end;  procedure  TForm1.AddSeparatorActionExecute(Sender: Tobject); begin  ActionManager1.DeleteActionItems(ActionI); end;  procedure  TForm1.AddSeparatorActionExecute(Sender: Tobject); var  Item: TActionClientItem; begin  Item: TActionClientItem; legin  Item: TActionClientItem;  Item: TActionClientItem;	• •	
ActionManager1.AddSeparator(Ite m); end;  procedure TForm1.AddCategoryActionExecute (Sender: TObject); var Item: TActionClientItem; begin Item: Gend;  procedure TForm1.AddActionActionExecute(Sender: TObject); if Assigned(Item) then  ActionManager1.AddCategory('Edit', Item); end;  procedure TForm1.AddActionActionExecute(Sender: TObject); var Item: TActionClientItem; begin Item: = ActionManager1.FindItemByCaption('&ActionManager1.FindItemByCaption('&ActionManager1.FindItemByCaption('ActionManager1.FindItemByCapti		,
ActionManager1.AddSeparator(Ite m); end;  procedure  TForm1.AddCategoryActionExecute (Sender: TObject); var  Item: TActionClientItem; begin  Item: ActionManager1.FindItemByCaption('&test'); if Assigned(Item) then  ActionManager1.AddCategory('Edit', Item); end;  procedure  TForm1.AddActionActionExecute(Sender: TObject); var  Item: TActionClientItem; begin  Item: ActionManager1.FindItemByCaption('&ActionI'); if Assigned(Item) then  ActionManager1.FindItemByCaption('&ActionI'); if Assigned(Item) then  ActionManager1.FindItemByCaption('&ActionI'); if Assigned(Item) then  ActionManager1.DisableBtnClick(Sender: TObject);  procedure  TForm1.DisableBtnClick(Sender: TObject);  procedure  TForm1.DisableBtnClick(Sender: TObject);  procedure  TForm1.DisableBtnClick(Sender: TObject);  procedure  TForm1.DeleteItemsActionExecute(Sender: TObject);  procedure  [\$R *.dfm\$}  procedure  TForm1.DeleteItemsActionExecute(Sender: TObject);  procedure  TForm1.AddActionActionExecute(Sender: TObject);  procedure  TForm1.AddActionActionExecute(Sender: TObject);  procedure  TForm1.AddActionActionExecute(Sender: TObject);  procedure  [\$R *.dfm\$}  ActionManager1.DeleteActionItems([Action1]);  end;  actionManager1.DeleteActionItems([Action1]);  end;  actionList1: TActionExecute(Sender: TObject);  procedure  : TObject);  procedure  [\$R *.dfm}  ActionManager1.DeleteActionItems([Action1]);  end;  actionManager1.DeleteActionExecute(Sender: TObject);  var  Item: TActionClientItem;  begin	if Assigned(Item) then	,
m); end; procedure TForm1.AddCategoryActionExecute (Sender: TObject); var Item: TActionClicntItem; begin ActionManager1.FindItemByCaptio n('&test'); if Assigned(Item) then  procedure TForm1.AddActionActionExecute(Sender: TObject); end;  procedure TForm1.AddActionActionExecute(Sender: TObject); procedure AddCategoryActionExecute(Sender: TObject); procedure AddCategoryActionExecute(Sender: TObject); procedure AddCategoryActionExecute(Sender: TObject); procedure AddActionActionExecute(Sender: TObject); procedure  ActionManager1.AddActionExecute(Sender: TObject); procedure  FForm1.AddActionActionExecute(Sender: TObject); procedure  Form1.AddActionActionExecute(Sender: TObject); procedure  Form1.Tobject); procedure  AddCategoryActionExecute(Sender: TObject); procedure  BisableBtnClick(Sender: TObject); private {Private declarations} end;  var  Item: TActionClientItem; begin  ActionManager1.FindItemByCaption('&Action1'); if Assigned(Item) then  procedure TForm1.DeleteItemsActionExecute(Sender: Tobject); begin  ActionManager1.DeleteActionItems( [Action1]); end;  procedure TForm1.AddSeparatorActionExecute(Sender: TObject); var Item: TActionClientItem; begin  Item: TActionClientItems(Sender: TObject); var Item: TActionClientItem;		DisableBtn: TButton;
end;  procedure TForm1.AddCategoryActionExecute (Sender: TObject); var  Item: TActionClientItem; begin  ActionManager1.FindItemByCaptio n('&test'); var  Item: TActionClientItem; begin  procedure  AddCategoryActionExecute(Sender: TObject); procedure AddCategoryActionExecute(Sender: TObject); procedure AddCategoryActionExecute(Sender: TObject); procedure AddCategoryActionExecute(Sender: TObject); procedure AddCategoryActionExecute(Sender: TObject); procedure AddActionActionExecute(Sender: TObject); procedure  Item: TForm1.AddActionExecute(Sender: TObject); private Private declarations } public Public declarations } end;  var  Form1: TForm1;  send;  implementation  SR *.dfm}  implementation  ActionManager1.FindItemByCaptio n('&ActionI'); if Assigned(Item) then  ActionManager1.AddAction(EditCutl, Item); end;  procedure TForm1.DeleteItemsActionExecute(Sender: TObject); begin  ActionManager1.DeleteActionItems([ActionI]); end;  procedure TForm1.AddSeparatorActionExecute(Sender: TObject); var  Item: TActionClientItem; begin Item: = ActionManager1.FindItemByCaptio n('&test');	ActionManager1.AddSeparator(Ite	ActionList1: TActionList;
procedure TForm1.AddCategoryActionExecute (Sender: TObject); var Item: TActionClientItem; begin ActionManager1.AddCategory('Edit', Item); end; procedure ActionManager1.FindItemByCaption ('&test'); var Item: TActionClientItem; begin Item: TActionClientItem; begin Item: = ActionManager1.FindItemByCaption ('&Action1'); if Assigned(Item) then  ActionManager1.FindItemByCaption ('&Action1'); if Assigned(Item) then  ActionManager1.FindItemByCaption ('&Action1'); if Assigned(Item) then  ActionManager1.AddAction(EditCut1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); begin  ActionManager1.AddAction(EditCut1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); begin  ActionManager1.DeleteActionItems( [Action1]); end;  procedure TForm1.AddSeparatorActionExecute( Sender: TObject); begin  ActionManager1.DeleteActionItems( [Action1]); end;  procedure TForm1.AddSeparatorActionExecute( Sender: TObject); begin  ActionManager1.DeleteActionItems( [Action1]); end;  procedure TForm1.AddSeparatorActionExecute( Sender: TObject); begin  ActionManager1.DeleteActionItems( [Action1]); end;  var Item: TActionClientItem; begin Item: TActionClientItem;	m);	EditDelete2: TEditDelete;
TObject);   Tobject);   Tobject);   Var	end;	procedure
TForm1.AddCategoryActionExecute (Sender: TObject); var  Item: TActionClientItem; begin  Item: = ActionManager1.FindItemByCaptio n('&test'); if Assigned(Item) then  ActionManager1.AddCategory('Edit ', Item); end;  procedure TForm1.AddActionActionExecute(Sender: TObject); var Item: TActionClientItem; begin Item: = ActionManager1.FindItemByCaptio n('&ActionI'); if Assigned(Item) then  ActionManager1.AddAction(EditCu t1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); begin  ActionManager1.AddAction(EditCu t1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); begin  Item: TActionClientItem; begin  Item: Tobject); var Item: TActionClientItem; begin  Item: Tobject); var Item: TActionClientItem; begin  Item: TActionClientItem;  Item: TActionClientItem;		DeleteItemsActionExecute(Sender:
TForm1.AddCategoryActionExecute (Sender: TObject);   var   Item: TActionClientItem;   begin   Item :=   AddCategoryActionExecute (Sender: TObject);   procedure   AddCationActionExecute (Sender: TObject);   procedure   DisableBtnClick (Sender: TObject);   private declarations }   public   { Public declarations }   end;	procedure	TObject);
Sender: TObject); var   Tobject);   Titem: TActionClientItem;   Degin   Tobject);   procedure   Tobj	TForm1.AddCategoryActionExecute	procedure
Item: TActionClientItem;   begin	(Sender: TObject);	•
Item: TActionClientItem; begin  Item := ActionManager1.FindItemByCaptio n('&test'); if Assigned(Item) then  ActionManager1.AddCategory('Edit ', Item); end;  procedure TForm1.AddActionActionExecute(Sender: TObject); var Item: TActionClientItem; begin Item := ActionManager1.FindItemByCaptio n('&Action1'); if Assigned(Item) then  ActionManager1.AddAction(EditCu t1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); begin  ActionManager1.AddAction(EditCu t1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); begin  ActionManager1.DeleteActionItems(	var	
Degin   Item :=   AddCategoryActionExecute(Sender: TObject);   procedure   AddCategory('Edit ', Item);   end;   procedure   Tobject);   private   {Private declarations }   public   {Public declarations }   end;	Item: TActionClientItem;	• • • • • • • • • • • • • • • • • • • •
Item := ActionManager1.AddCategory('Edit ', Item); end;  Procedure Trom1.AddActionActionExecute(Sender: TObject); var Item: TActionClientItem; begin Item := ActionManager1.FindItemByCaption('&ActionI'); if Assigned(Item) then  ActionManager1.FindItemByCaption('&ActionI'); if Assigned(Item) then  ActionManager1.AddAction(EditCut1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); begin  ActionManager1.AddAction(EditCut1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); begin  ActionManager1.DeleteActionItems([ActionI]); end;  procedure TForm1.AddSeparatorActionExecute(Sender: TObject); var Item: TActionClientItem; begin Item := ActionManager1.FindItemByCaption('&test');  Item: TActionClientItem; Item: TActionClientItem; Item: TActionClientItem; Item: TActionClientItem;	begin	-
ActionManager1.FindItemByCaptio n('&test'); if Assigned(Item) then  ActionManager1.AddCategory('Edit ', Item); end;  procedure TForm1.AddActionActionExecute(Sender: TObject); var Item: TActionClientItem; begin Item := ActionManager1.FindItemByCaptio n('&Action1'); if Assigned(Item) then  ActionManager1.AddAction(EditCut1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); var Item: TActionClientItem; begin  procedure ActionManager1.AddAction(EditCut1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); begin  ActionManager1.DeleteActionItems( [Action1]); end;  procedure TForm1.AddSeparatorActionExecute(Sender: TObject); var Item: TActionClientItem; begin  Item: TActionClientItem;		,
n('&test'); if Assigned(Item) then  ActionManager1.AddCategory('Edit', Item); end;  procedure TForm1.AddActionActionExecute(Sender: TObject); var Item: TActionClientItem; begin Item:= ActionManager1.FindItemByCaption('&Action1'); if Assigned(Item) then  ActionManager1.AddAction(EditCut1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); var Item: TActionClientItem; begin  procedure TForm1.DisableBtnClick(Sender: TObject); var Item: TActionClientItem; begin Item:= ActionManager1.FindItemByCaption('&test');  var Item: TActionClientItem; begin Item:= ActionManager1.FindItemByCaption('&test');  Item: TActionClientItem;		* .
if Assigned(Item) then  ActionManager1.AddCategory('Edit ', Item); end;  procedure  Fform1.AddActionActionExecute(Sender: TObject); var  Item: TActionClientItem; begin  ActionManager1.FindItemByCaption('&ActionI'); if Assigned(Item) then  ActionManager1.AddAction(EditCut1, Item); end;  procedure  Fform1.DisableBtnClick(Sender: TObject); var  Item: TActionClientItem; begin  procedure  Form1.DisableBtnClick(Sender: TObject); var  Item: TActionClientItem; begin  Item: TActionClientItem; Item: TActionClientItem; Item: TActionClientItem; Item: TActionClientItem;		-
ActionManager1.AddCategory('Edit ', Item); end;  procedure TForm1.AddActionActionExecute(S ender: TObject); var Item: TActionClientItem; begin Item := ActionManager1.FindItemByCaptio n('&Action1'); if Assigned(Item) then  ActionManager1.AddAction(EditCu t1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); var Item: TActionClientItem; begin  ActionManager1.AddAction(EditCu t1, Item); end;  ActionManager1.DeleteActionItems( [Action1]); end;  ActionManager1.DeleteActionItems( [Action1]); end;  Trorm1.AddSeparatorActionExecute( Sender: TObject); var Item: TActionClientItem; begin  Item: TActionClientItem;		`
ActionManager1.AddCategory('Edit ', Item); end;  procedure  TForm1.AddActionActionExecute(S ender: TObject); var  Item: TActionClientItem; begin  Item :=  ActionManager1.FindItemByCaptio n('&Action1'); if Assigned(Item) then  ActionManager1.AddAction(EditCu t1, Item); end;  procedure  TForm1.DisableBtnClick(Sender: TObject); var  Item: TActionClientItem; begin  ActionManager1.AddAction(EditCu t1, Item); end;  ActionManager1.DeleteActionItems( [Action1]); end;  procedure  TForm1.DisableBtnClick(Sender: TObject); var  Item: TActionClientItem; begin  Item: TActionClientItem; begin  Item: TActionClientItem; Item: TAct		
ry. Item); end; end;  procedure TForm1.AddActionActionExecute(Sender: TObject); var Item: TActionClientItem; begin Item := ActionManager1.FindItemByCaption('&Action1'); if Assigned(Item) then  ActionManager1.AddAction(EditCut1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); var Item: TActionClientItem; begin Item: TActionClientItem; Item: TActionClientItem; Item: TActionClientItem; Item: TActionClientItem;	ActionManager1.AddCategory('Edit	-
end;  procedure TForm1.AddActionExecute(S ender: TObject); var  Item: TActionClientItem; begin  Item := ActionManager1.FindItemByCaptio n('&Action1'); if Assigned(Item) then  ActionManager1.AddAction(EditCu t1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); var  Item: TActionClientItem; begin  procedure TForm1.DisableBtnClick(Sender: TObject); var  Item: TActionClientItem; begin  Item: TActionClientItem;	0 0,	
procedure TForm1.AddActionActionExecute(S ender: TObject); var Item: TActionClientItem; begin Item := ActionManager1.FindItemByCaption('&Action1'); if Assigned(Item) then  ActionManager1.AddAction(EditCut1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); var Item: TActionClientItem; begin Item: TActionClientItem;		*
TForm1.AddActionActionExecute(Sender: TObject);   var	- Chu,	
TForm1.AddActionActionExecute(S ender: TObject);  var  Item: TActionClientItem; begin  Item := ActionManager1.FindItemByCaptio n('&Action1'); if Assigned(Item) then  ActionManager1.AddAction(EditCu t1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); var  Item: TActionClientItem; begin  Item: TActionClientItem; begin  Item: = ActionManager1.FindItemByCaptio n('&test');  Item: TActionClientItem;	nrocedure	
ender: TObject);  var  Item: TActionClientItem; begin  Item := ActionManager1.FindItemByCaptio n('&Action1'); if Assigned(Item) then  ActionManager1.AddAction(EditCu t1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); var  Item: TActionClientItem; begin  Item: TActionClientItem; begin  Item: TActionManager1.FindItemByCaptio n('&test');  var  Item: TActionClientItem;  Item: TActionClientItem;  Item: TActionClientItem;  Item: TActionClientItem;  Item: TActionClientItem;  Item: TActionClientItem;		
varvarItem: TActionClientItem;Form1: TForm1;beginimplementationActionManager1.FindItemByCaptio n('&Action1'); if Assigned(Item) then[\$R *.dfm}ActionManager1.AddAction(EditCu t1, Item); end;procedure TForm1.DeleteItemsActionExecute( Sender: TObject); beginprocedure TForm1.DisableBtnClick(Sender: TObject); varActionManager1.DeleteActionItems( [Action1]); end;Item: TActionClientItem; beginprocedure TForm1.AddSeparatorActionExecut e(Sender: TObject); varItem: = ActionManager1.FindItemByCaptio n('&test');Item: TActionClientItem;		enu,
Item: TActionClientItem; begin  Item := ActionManager1.FindItemByCaptio n('&Action1'); if Assigned(Item) then  ActionManager1.AddAction(EditCu t1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); var Item: TActionClientItem; begin Item: TActionClientItem; begin  Item: TActionManager1.FindItemByCaptio n('&test');  Form1: TForm1;  Form1: TForm1;  Form1: TForm1;  Form1: TForm1;  Form1: TForm1;  ActionManager1.FindItemByCaptio implementation  ActionManager1 Form1.DeleteItemsActionExecute( Sender: TObject); end;  Form1: TForm1;  Form1: TForm1;  Implementation  ActionManager1.FindItemSyCaptio TForm1.DeleteItemsActionExecute( Sender: TObject); end;  Item: TActionClientItem;  Item: TActionClientItem;	* * * * * * * * * * * * * * * * * * * *	Van
Item :=   ActionManager1.FindItemByCaptio n('&Action1');		
Item := ActionManager1.FindItemByCaptio n('&Action1'); if Assigned(Item) then  ActionManager1.AddAction(EditCu t1, Item); end;  procedure TForm1.DeleteItemsActionExecute( Sender: TObject); begin  ActionManager1.DeleteActionItems( [Action1]); end;  var  Item: TActionClientItem; begin  Item:= ActionManager1.FindItemByCaptio n('&test');  implementation  {\$R *.dfm}}  ActionManager1.DeleteItemsActionExecute( Sender: TObject); end;  procedure TForm1.AddSeparatorActionExecute( Sender: TObject); var Item: TActionClientItem;		FORMI: 1FORMI;
ActionManager1.FindItemByCaptio n('&Action1'); if Assigned(Item) then  ActionManager1.AddAction(EditCu t1, Item); end;  procedure TForm1.DeleteItemsActionExecute( Sender: TObject); begin  ActionManager1.DeleteActionItems( [Action1]); end;  var  Item: TActionClientItem; begin  Item:= ActionManager1.FindItemByCaptio n('&test');  Item: TActionClientItem;		:
n('&Action1');  if Assigned(Item) then  ActionManager1.AddAction(EditCut1, Item); end;  procedure TForm1.DeleteItemsActionExecute(Sender: TObject); begin  ActionManager1.DeleteActionItems([Action1]); end;  ActionManager1.DeleteActionItems([Action1]); end;  rocedure TForm1.AddSeparatorActionExecute(Sender: TObject); var Item: TActionClientItem; begin  Item: TActionManager1.FindItemByCaption('&test');  Item: TActionClientItem;		implementation
if Assigned(Item) then  ActionManager1.AddAction(EditCut1, Item); end;  procedure TForm1.DeleteItemsActionExecute(Sender: TObject); begin  ActionManager1.DeleteActionItems([Action1]); end;  var Item: TActionClientItem; begin Item:= ActionManager1.FindItemByCaption('&test');  Item: TActionClientItem; Item: TActionClientItem; Item: TActionClientItem; Item: TActionClientItem;		(OD + 10 )
ActionManager1.AddAction(EditCut1, Item); end;  procedure TForm1.DeleteItemsActionExecute(Sender: TObject); begin  ActionManager1.DeleteActionItems([Action1]); end;  var Item: TActionClientItem; begin  Item: TActionClientItem; begin  TForm1.AddSeparatorActionExecute(Sender: TObject); var Item: TActionManager1.FindItemByCaption('&test');  Item: TActionClientItem;		{\$R *.dtm}
ActionManager1.AddAction(EditCu t1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); var  Item: TActionClientItem; begin  Item:= ActionManager1.FindItemByCaptio n('&test');  TForm1.DeleteItemsActionExecute(Sender: TObject); begin  ActionManager1.DeleteActionItems([Action1]); end;  TForm1.DeleteItemsActionExecute(Sender: TObject); begin  ActionManager1.DeleteActionItems([Action1]); end;  TForm1.DeleteItemsActionExecute(Sender: TObject); begin	n Assigned(item) then	_
t1, Item); end;  procedure TForm1.DisableBtnClick(Sender: TObject); var  Item: TActionClientItem; begin  Item := ActionManager1.DeleteActionItems( [Action1]); end;  procedure TForm1.AddSeparatorActionExecut e(Sender: TObject); var  Item: TActionClientItem;  tem: TActionClientItem;	ActionMonogoul Add Action (Editor	1 =
end;  procedure TForm1.DisableBtnClick(Sender: TObject); var Item: TActionClientItem; begin  Item:= ActionManager1.DeleteActionItems( [Action1]); end;  procedure TForm1.AddSeparatorActionExecut e(Sender: TObject); var Item: TActionClientItem;	Č	· ·
procedure TForm1.DisableBtnClick(Sender: TObject); var  Item: TActionClientItem; begin Item := ActionManager1.DeleteActionItems( [Action1]); end;  procedure TForm1.AddSeparatorActionExecut e(Sender: TObject); var  Item: TActionClientItem;		<b>9</b> 77
TForm1.DisableBtnClick(Sender: TObject);  var  Item: TActionClientItem; begin Item := ActionManager1.FindItemByCaptio n('&test');  Item: TActionClientItem;  Item: TActionClientItem;  Item: TActionClientItem;	ena;	begin
TForm1.DisableBtnClick(Sender: TObject);  var  Item: TActionClientItem; begin Item := ActionManager1.FindItemByCaptio n('&test');  Item: TActionClientItem;  Item: TActionClientItem;  Item: TActionClientItem;		
TObject); var  Item: TActionClientItem; begin  Item:= ActionManager1.FindItemByCaptio n('&test');  rend;  procedure TForm1.AddSeparatorActionExecut e(Sender: TObject); var Item: TActionClientItem;	-	
varItem: TActionClientItem;procedurebeginTForm1.AddSeparatorActionExecutItem :=e(Sender: TObject);ActionManager1.FindItemByCaptiovarn('&test');Item: TActionClientItem;		[Action1]);
Item: TActionClientItem;procedurebeginTForm1.AddSeparatorActionExecutItem :=e(Sender: TObject);ActionManager1.FindItemByCaptiovarn('&test');Item: TActionClientItem;	¥ //	end;
begin  Item := ActionManager1.FindItemByCaptio n('&test');  TForm1.AddSeparatorActionExecut e(Sender: TObject); var  Item: TActionClientItem;		
Item := ActionManager1.FindItemByCaptio n('&test');e(Sender: TObject); varItem: TActionClientItem;		procedure
ActionManager1.FindItemByCaptio n('&test');  Item: TActionClientItem;		TForm1.AddSeparatorActionExecut
n('&test');  Item: TActionClientItem;		<u>-</u>
	ActionManager1.FindItemByCaptio	var
	n('&test');	Item: TActionClientItem;
	if Assigned(Item) then	begin

end.

15660791
Font.Charset = ANSI CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'Tahoma'
Font.Style = []
Spacing = 3
end
object Memo1: TMemo
Left = 0
Top = 60
Width = 490
Height = 225
Align = alClient
TabOrder = 1
end
object Panel1: TPanel
Left = 0
Top = 24
Width = 490
Height = 36
Align = alTop
BevelOuter = bvNone
TabOrder = 2
object Button4: TButton
Left = 260
Top = 6
Width = 102
Height = 25
Action = DeleteItemsAction
TabOrder = 3
end
object Button1: TButton
Left = 171
Top = 6
Width = 82
Height = 25
Action = AddSeparatorAction
TabOrder = 2
end
object Button3: TButton
Left = 8
Top = 6
Width = 75
Height = 25

Item.Control.Enabled := not
Item.Control.Enabled;
end;

ObjectS
الملف النصي لبرنامج
(DynaActionBands)
إعداد
علاء الدين اللباد
واتف ۷۹۴۵۷۵۴۷۱
DynaActionBands
object Form1: TForm1
Left = -63
Top = 117
Width = 498
Height = 338
Caption = 'ActionManager Helper
Demo'
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = False
ShowHint = True
PixelsPerInch = 96
TextHeight = 13
object ActionMainMenuBar1:
TActionMainMenuBar
Left = 0
Top = 0
Width = 490
Height = 24
UseSystemFont = False
ActionManager = ActionManager1
Caption = 'ActionMainMenuBar1'
ColorMap.HighlightColor =
15660791
ColorMap.BtnSelectedColor =
clBtnFace
ColorMap.UnusedColor =

end>
LinkedActionLists = <
item
ActionList = ActionList1
Caption = 'ActionList1'
end>
Left = 61
Top = 73
StyleName = 'XP Style'
object Action1: TAction
Category = 'test'
Caption = 'Action1'
Enabled = False
end
object Action2: TAction
Category = 'Test'
Caption = 'Action2'
Enabled = False
end
object EditCut1: TEditCut
Category = 'Edit'
Caption = 'Cu&t'
Enabled = False
Hint = 'Cut Cuts the selection
and puts it on the Clipboard'
ImageIndex = 0
<b>ShortCut = 16472</b>
end
object EditCopy1: TEditCopy
Category = 'Edit'
Caption = '&Copy'
Enabled = False
Hint = 'Copy Copies the selection
and puts it on the Clipboard'
ImageIndex = 1
ShortCut = 16451
end
object EditPaste1: TEditPaste
Category = 'Edit'
Caption = '&Paste'
Enabled = False
Hint = 'Paste Inserts Clipboard
contents'
ImageIndex = 2
ShortCut = 16470
end
object EditSelectAll1:
TEditSelectAll

Action = AddCategoryAction
TabOrder = 0
end
object Button2: TButton
Left = 89
Top = 6
Width = 75
Height = 25
Action = AddActionAction
TabOrder = 1
end
object DisableBtn: TButton
Left = 372
Top = 6
Width = 112
Height = 25
Caption = 'Disable Menu Button'
TabOrder = 4
OnClick = DisableBtnClick
end
end
object StatusBar1: TStatusBar
Left = 0
Top = 285
Width = 490
Height = 19
AutoHint = True
Panels = <>
end
object ActionManager1:
TActionManager
ActionBars = <
item
Items = <
item
Items = <
item
Action = Action1
Caption = '&Action1'
end
item
Action = Action2
Caption = 'A&ction2'
end>
Caption = '&test'
end>
ActionBar =
ActionMainMenuBar1
ACHOHMAHIMEHUDAFI

object DeleteItemsAction: TAction
Category = 'Buttons'
Caption = 'Delete Action Items'
Hint = 'Deletes all items attached
to Action1'
OnExecute =
DeleteItemsActionExecute
end
end
object ActionList1: TActionList
Left = 194
Top = 74
object EditDelete2: TEditDelete
Category = 'Edit'
Caption = '&Delete'
Hint = 'Delete Erases the
selection'
ImageIndex = 5
ShortCut = 46
end
end
end

Category = 'Edit'
Caption = 'Select &All'
Enabled = False
Hint = 'Select All Selects the
entire document'
<b>ShortCut = 16449</b>
end
object EditUndo1: TEditUndo
Category = 'Edit'
Caption = '&Undo'
Enabled = False
Hint = 'Undo Reverts the last
action'
ImageIndex = 3
ShortCut = 16474
end
object AddCategoryAction:
TAction
Category = 'Buttons'
Caption = 'Add Category'
Hint =
'Adds the "Edit" Category to
the menu after the item with the cap'
+
'tion ''&Test'''
OnExecute =
<b>AddCategoryActionExecute</b>
end
object AddActionAction: TAction
Category = 'Buttons'
Caption = 'Add Action'
Hint =
'Adds the EditCut action after
the item with the caption "&Action"
+
'1'''
OnExecute =
AddActionActionExecute
end
object AddSeparatorAction:
TAction  Cotogory = !Puttons!
Category = 'Buttons'
Caption = 'Add Separator'
Hint = 'Adds a separator after
the item that is connected to
Action1'
OnExecute =
AddSeparatorActionExecute
end

Controls, Forms,	program ABMI
Dialogs, ActnList, StdActns,	
ToolWin, ActnMan, ActnCtrls,	uses
ActnMenus,	Forms,
StdCtrls, ImgList, ComCtrls,	main in 'main,
ExtActns, XPStyleActnCtrls,	
BandActn,	{\$R *.res}
StdStyleActnCtrls;	,
	begin
type	Application.In
TForm1 = class(TForm)	Application.C
ActionManager1:	Form1);
TActionManager;	Application.R
ActionMainMenuBar1:	end.
TActionMainMenuBar;	
Action1: TAction;	{
Action2: TAction;	NOTE: To ad
Action3: TAction;	reopen menu si
ReopenActionList1: TActionList;	actions to the
Action4: TAction;	ReopenAc
Action5: TAction;	OnExecute ever
Action6: TAction;	ReopenActionE
Action7: TAction;	event.
Action8: TAction;	
ActionToolBar1:	Also, in or
TActionToolBar;	file to be saved
ImageList1: TImageList;	close this a
FileOpen1: TFileOpen;	one time. So, to
EditCut1: TEditCut;	items worl
EditCopy1: TEditCopy;	application and
EditPaste1: TEditPaste;	then use
EditSelectAll1: TEditSelectAll;	it normall
EditUndo1: TEditUndo;	
EditDelete1: TEditDelete;	ISSUE: If you
RichEditBold1: TRichEditBold;	customizations
RichEditItalic1: TRichEditItalic;	there is potentia
RichEditUnderline1:	for the us
TRichEditUnderline;	Reopen menu it
RichEditStrikeOut1:	toolbar button
TRichEditStrikeOut;	which this
RichEditBullets1:	NOT take into a
TRichEditBullets;	}
RichEditAlignLeft1:	unit main;
TRichEditAlignLeft;	unit main,
RichEditAlignRight1:	interface
8 8	interrace
TRichEditAlignRight;	11505
RichEditAlignCenter1:	Windows Mos
TRichEditAlignCenter;	Windows, Mes Variants, Class
SearchFind1: TSearchFind;	v ai iaiits, Ciass

program ABMRU;  uses Forms, main in 'main.pas' {Form1};  {\$R *.res}  begin Application.Initialize; Application.CreateForm(TForm1, Form1); Application.Run; end.  { NOTE: To add more items to the reopen menu simply add more actions to the
Forms, main in 'main.pas' {Form1};  {\$R *.res}  begin Application.Initialize; Application.CreateForm(TForm1, Form1); Application.Run; end.  { NOTE: To add more items to the reopen menu simply add more actions to the
Forms, main in 'main.pas' {Form1};  {\$R *.res}  begin Application.Initialize; Application.CreateForm(TForm1, Form1); Application.Run; end.  { NOTE: To add more items to the reopen menu simply add more actions to the
main in 'main.pas' {Form1};  {\$R *.res}  begin Application.Initialize; Application.CreateForm(TForm1, Form1); Application.Run; end.  { NOTE: To add more items to the reopen menu simply add more actions to the
begin Application.Initialize; Application.CreateForm(TForm1, Form1); Application.Run; end.  { NOTE: To add more items to the reopen menu simply add more actions to the
begin Application.Initialize; Application.CreateForm(TForm1, Form1); Application.Run; end.  { NOTE: To add more items to the reopen menu simply add more actions to the
begin Application.Initialize; Application.CreateForm(TForm1, Form1); Application.Run; end.  { NOTE: To add more items to the reopen menu simply add more actions to the
Application.Initialize; Application.CreateForm(TForm1, Form1); Application.Run; end.  { NOTE: To add more items to the reopen menu simply add more actions to the
Application.Initialize; Application.CreateForm(TForm1, Form1); Application.Run; end.  { NOTE: To add more items to the reopen menu simply add more actions to the
Application.CreateForm(TForm1, Form1); Application.Run; end.  { NOTE: To add more items to the reopen menu simply add more actions to the
Application.Run; end.  {  NOTE: To add more items to the reopen menu simply add more actions to the
NOTE: To add more items to the reopen menu simply add more actions to the
NOTE: To add more items to the reopen menu simply add more actions to the
NOTE: To add more items to the reopen menu simply add more actions to the
NOTE: To add more items to the reopen menu simply add more actions to the
reopen menu simply add more actions to the
actions to the
ReopenActionList and set their
OnExecute event to the ReopenActionExecute
event.
event.
Also, in order for the settings
file to be saved you must run and
close this application at least
one time. So, to see the reopen
items working run the
application and shut it down first
then use
it normally.
ISSUE: If you allow
customizations in your application
there is potential
for the user to delete the
Reopen menu item or the Open toolbar button
which this application does
NOT take into account.
}
unit main;
· · · · · · · · · · · · · · · · · · ·
interface
_
uses
Windows, Messages, SysUtils,
Variants, Classes, Graphics,

begin	SearchFindNext1:
	TSearchFindNext;
// Find the Reopen item by looking	,
at the item caption	SearchReplace1: TSearchReplace;
if AClient is TActionClientItem	SearchFindFirst1:
then	TSearchFindFirst;
if Pos('Reopen',	FileSaveAs1: TFileSaveAs;
TActionClientItem(AClient).Caption	FilePrintSetup1: TFilePrintSetup;
) <> 0 then	FileRun1: TFileRun;
ReopenMenuItem := AClient as	FileExit1: TFileExit;
TActionClientItem	Action9: TAction;
end;	RichEdit1: TRichEdit;
	CustomizeActionBars1:
procedure	TCustomizeActionBars;
TForm1.FindOpenToolItem(AClient	procedure FormCreate(Sender:
: TActionClient);	TObject);
begin	procedure
// Find the Open item by looking at	FileOpen1Accept(Sender: TObject);
the item caption	procedure
if AClient is TActionClientItem	ReopenActionExecute(Sender:
then	TObject);
if Pos('Open',	private
TActionClientItem(AClient).Caption	•
) $\Leftrightarrow$ 0 then	{ Private declarations }
/	ReopenMenuItem:
OpenToolItem := AClient as	TActionClientItem;
TActionClientItem;	OpenToolItem:
end;	TActionClientItem;
	procedure
procedure	FindReopenMenuItem(AClient:
TForm1.FormCreate(Sender:	TActionClient);
TObject);	procedure
	FindOpenToolItem(AClient:
procedure	TActionClient);
SetupItemCaptions(AnItem:	procedure
TActionClientItem);	UpdateReopenItem(ReopenItem:
var	TActionClientItem);
I: Integer;	public
begin	{ Public declarations }
if Assigned(AnItem) then	end;
for I := 0 to AnItem.Items.Count	
- 1 do	var
	Form1: TForm1;
TCustomAction(ReopenActionList1.	, , , , , , , , , , , , , , , , , , ,
Actions[I]).Caption :=	implementation
	mprementation
Copy(AnItem.Items[I].Caption, 5,	(\$D * dfm)
MaxInt);	{\$R *.dfm}
end;	nua sa duna
VIII,	procedure
hogin	TForm1.FindReopenMenuItem(ACl
begin	ient: TActionClient);

RichEdit1.Align:=alClient;   // Find the Reopen menu item on the ActionMainMenu		
m);  UpdateReopenItem(OpenToolItem); end;  //then actually open the file  RichEdit1.Lines.LoadFromFile(File OpenI.Dialog.FileName); end;  // condition file file file file openI.Dialog.FileName); end;  // condition file file openI.Dialog.FileName); end;  // Add thew new filename to the beginning of the list and move other items down for 1:= ReopenActionList1.ActionCount - 1 downto 0 do     if I = 0 then  TCustomAction(ReopenActionList1. Actions[I]).Caption := FileOpen1.Dialog.FileName else  TCustomAction(ReopenActionList1. Actions[I - I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I - I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I - I]).Caption :=  fleopenItem.Items.Count < ReopenActionList1.ActionCount - I do		RichEdit1.Align := alClient;
UpdateReopenItem(OpenToolItem); end;  //then actually open the file  RichEdit1.Lines.LoadFromFile(File Open1.Dialog.FileName); end;  procedure TForm1.UpdateReopenItem(Reopen Item: TActionClientItem); var I: Integer; begin if ReopenItem = nil then exit; // Add thew new filename to the beginning of the list and move other items down for I := ReopenActionList1.ActionCount - 1 downto 0 do if I = 0 then  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I].Caption :=  TCustomAction(ReopenActionList1. Actions[I].Caption :=  TCustomAction(ReopenActionList1. Actions[I].Caption :=  TCustomAction(ReopenActionList1. Actions[I].Caption; // Add new items to the reopen item if necessary if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenActionList1.ActionCount then  ReopenItem.Items.Add; // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  ActionManager1.ActionBars.Iterate Clients(ActionManager1.ActionBars IfindReopenMenuItem); // Find the Reopen menu item on the ActionToolBar  ActionManager1.ActionBars.Iterate Clients(ActionManager1.ActionBars III.tems, FindOpenToolItem); // Set the captions (ReopenMenuIte m);  SetupItemCaptions(ReopenActionS(PopenToolItem); end;  I: Integer;  Found: Baraction Bars III.tems. FindOpenToolItem); // Set the captions (ReopenMenuIte m);  SetupItemCaptions(CopenToolItem); var  I: Integer;  Found: Baraction Bars III.tems. FindOpenToolItem); // Set the captions of the actions since they are used to open the file m);  SetupItemCaptions(CopenColien); // Items. FindOpenToolItem); // Set the captions of the actions since they are used to open the file m);  SetupItemCaptions(action List1. Actions[I].Caption:  I: Integer; Found: Baraction List1. Actions[I].Caption:  I: Integer; Found: Baraction List1. Actions[I].Caption: III. Inte	UpdateReopenItem(ReopenMenuIte	// Find the Reopen menu item on
end;  //then actually open the file  RichEdit1.Lines.LoadFromFile(File Open1.Dialog.FileName); end;  procedure TForm1.UpdateReopenItem(Reopen Item: TActionClientItem); var  I: Integer; begin if ReopenItem = nil then exit;  // Add thew new filename to the beginning of the list and move other items down  for I := ReopenActionList1.ActionCount - 1 downto 0 do  if I = 0 then  TCustomAction(ReopenActionList1. Actions[I]).Caption := FileOpen1.Dialog.FileName clse  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I] - 1]).Caption;  // Add new items to the reopen item if necessary  // ReopenActionList1.ActionCount then  ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the captions of the actions since they are used to open the file  SetupItemCaptions(ReopenMenuIte m);  SetupItemCaptions(OpenToolItem); end;  SetupItemCaptions(OpenToolItem); end;  I: Integer; Found: Boolean; begin  Found := False;  // If the filename is already in the list then do not add it again for I := 0 to ReopenActionList1.ActionCount - 1 do  if CompareText(TCustomAction(ReopenActionList1.Actions[I]).Caption, FileOpen1.Dialog.FileName) = 0 then  begin  Found := True; break; end; if not Found then	m);	the ActionMainMenu
end;  //then actually open the file  RichEdit1.Lines.LoadFromFile(File Open1.Dialog.FileName); end;  procedure TForm1.UpdateReopenItem(Reopen Item: TActionClientItem); var  I: Integer; begin if ReopenItem = nil then exit;  // Add thew new filename to the beginning of the list and move other items down  for I := ReopenActionList1.ActionCount - 1 downto 0 do  if I = 0 then  TCustomAction(ReopenActionList1. Actions[I]).Caption := FileOpen1.Dialog.FileName clse  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I] - 1]).Caption;  // Add new items to the reopen item if necessary  // ReopenActionList1.ActionCount then  ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the captions of the actions since they are used to open the file  SetupItemCaptions(ReopenMenuIte m);  SetupItemCaptions(OpenToolItem); end;  SetupItemCaptions(OpenToolItem); end;  I: Integer; Found: Boolean; begin  Found := False;  // If the filename is already in the list then do not add it again for I := 0 to ReopenActionList1.ActionCount - 1 do  if CompareText(TCustomAction(ReopenActionList1.Actions[I]).Caption, FileOpen1.Dialog.FileName) = 0 then  begin  Found := True; break; end; if not Found then		
//then actually open the file  RichEdit1.Lines.LoadFromFile(File Open1.Dialog.FileName); end;  procedure TForm1.UpdateReopenItem(Reopen Item: TActionClientItem); var  I: Integer; begin if ReopenItem = nil then exit; // Add thew new filename to the beginning of the list and move other items down for 1 := ReopenActionList1.ActionCount - 1 downto 0 do     if I = 0 then  TCustomAction(ReopenActionList1. Actions[I]).Caption := FileOpen1.Dialog.FileName     else  TCustomAction(ReopenActionList1. Actions[I]).Caption := FileOpen1.Dialog.FileName) = 0 then  ReopenActionList1.ActionCount then  ReopenActionList1.ActionSuption, FileOpen1.Dialog.FileName) = 0 then  begin Found := True; break; cnd; if not Found then  begin	UpdateReopenItem(OpenToolItem);	ActionManager1.ActionBars.Iterate
FindReopenMenuItem);   RichEdit1.Lines.LoadFromFile(File Open1.Dialog.FileName);   end;   procedure   TForm1.UpdateReopenItem(Reopen Item: TActionClientItem);   var	end;	Clients(ActionManager1.ActionBars
FindReopenMenuItem);   RichEdit1.Lines.LoadFromFile(File Open1.Dialog.FileName);   end;   procedure   TForm1.UpdateReopenItem(Reopen Item: TActionClientItem);   var	//then actually open the file	[0].Items,
RichEdit1.Lines.LoadFromFile(File Open1.Dialog.FileName); end;   ActionManager1.ActionBars.Iterate Clients(ActionManager1.ActionBars II].Items,	* 1	FindReopenMenuItem);
the ActionToolBar    Compart	RichEdit1.Lines.LoadFromFile(File	
end;  procedure TForm1.UpdateReopenItem(Reopen Item: TActionClientItem); var  I: Integer; begin if ReopenItem = nil then exit;  // Add thew new filename to the beginning of the list and move other items down for I := ReopenActionList1.ActionCount - 1 downto 0 do     if I = 0 then  TCustomAction(ReopenActionList1. Actions[I].Caption := FileOpen1.Dialog.FileName else  TCustomAction(ReopenActionList1. Actions[I].Caption :=  TCustomAction(ReopenActionList1. Actions[I].Caption :=  TCustomAction(ReopenActionList1. Actions[I].Caption :=  TCustomAction(ReopenActionList1. Actions[I].Caption :=  ITCustomAction(ReopenActionList1. Actions[I].Caption :=  TCustomAction(ReopenActionList1. Actions[I].Caption :=  ITCustomAction(ReopenActionList1. Actions[I].Caption :=  ITcustomAction(ReopenActionList1. Actions[I].Caption :=  If ReopenItem.Items.Count <  ReopenActionList1.ActionCount - 1 do  if CompareText(TCustomAction(ReopenActionList1.Actions[I]).Caption,  FileOpen1.Dialog.FileName) = 0 then  begin  Found := True;  break; end; if not Found then  begin		_
ActionManager1.ActionBars.Iterate Clients(ActionManager1.ActionBars II: Integer; begin  if ReopenItem = nil then exit; // Add thew new filename to the beginning of the list and move other items down for I := ReopenActionList1.ActionCount - 1 downto 0 do     if I = 0 then  TCustomAction(ReopenActionList1. Actions[I]).Caption := FileOpen1.Dialog.FileName     else  TCustomAction(ReopenActionList1. Actions[I - 1]).Caption :=  TCustomAction(ReopenActionList1. Actions[I - 1]).Caption :=  TCustomAction(ReopenActionList1. Actions[I - 1]).Caption; // Add new items to the reopen item if necessary if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add; // Set the item captions by appending a number for use as the shortcut // This change will cause them to be streamed which allows us to store	1 0 //	
Transfer		ActionManager1.ActionBars.Iterate
TForm1.UpdateReopenItem(Reopen Item: TActionClientItem); var  I: Integer; begin  if ReopenItem = nil then exit;  // Add thew new filename to the beginning of the list and move other items down  for I := ReopenActionList1.ActionCount - 1 downto 0 do     if I = 0 then  TCustomAction(ReopenActionList1. Actions[I]).Caption := FileOpen1.Dialog.FileName     else  TCustomAction(ReopenActionList1. Actions[I]).Caption := TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I]).Caption; // Add new items to the reopen item if necessary if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add; // Set the captions of the actions since they are used to open the file  SetupItemCaptions(OpenToolItem); end;  SetupItemCaptions(OpenToolItem); end;  Found: Boolean; begin  Found: False; // If the filename is already in the list then do not add it again for I := 0 to ReopenActionList1.ActionCount - 1 do  if CompareText(TCustomAction(ReopenActionList1.Actions[I]).Caption, FileOpen1.Dialog.FileName) = 0 then  begin  Found: = True; break; end; if not Found then	nrocedure	
Item: TActionClientItem); var  I: Integer; begin if ReopenItem = nil then exit;  // Add thew new filename to the beginning of the list and move other items down for I := ReopenActionList1.ActionCount - 1 downto 0 do if I = 0 then  TCustomAction(ReopenActionList1. Actions[I]).Caption := FlieOpen1.Dialog.FileName else  TCustomAction(ReopenActionList1. Actions[I]).Caption := TCustomAction(ReopenActionList1. Actions[I-1]).Caption :=  TCustomAction(ReopenActionList1. Actions[I-1]).Caption; // Add new items to the reopen item if necessary if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenActionList1.ActionCount then  ReopenItem.Items.Add; // Set the captions of the actions since they are used to open the file  SetupItemCaptions(PeopenMenuIte m);  SetupItemCaptions(OpenToolItem); end;  Pround: FileOpen1.FileOpen1Accept(Sender: TObject); var  I: Integer; Found: Boolean; begin  Found: = False; // If the filename is already in the list then do not add it again for I := 0 to ReopenActionList1.ActionCount - 1 do  if CompareText(TCustomAction(ReopenActionList1.ActionList1.ActionSII).Caption, FileOpen1.Dialog.FileName) = 0 then begin  Found: = True; break; end; if not Found then begin	-	` S
Var   I: Integer;   begin   if ReopenItem = nil then exit;   // Add thew new filename to the beginning of the list and move other items down   for I :=   ReopenActionList1.ActionCount - 1 downto 0 do   if I = 0 then		
I: Integer; begin  if ReopenItem = nil then exit;  // Add thew new filename to the beginning of the list and move other items down  for I := ReopenActionList1.ActionCount - 1 downto 0 do     if I = 0 then  TCustomAction(ReopenActionList1. Actions[I]).Caption := FileOpen1.Dialog.FileName     else  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  iTCustomAction(ReopenActionList1. Actions[I]).Caption;  // Add new items to the reopen item if necessary  if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  setupItemCaptions(ReopenMenuIte m);  setupItemCaptions(PeopenMenuIte m);  end;  It Integer;  Found := False;  // If the filename is already in the list then do not add it again for I := 0 to  ReopenActionList1.ActionCount - 1 do  ReopenActionList1.ActionCount - 1 do  ReopenItem.Items.Count <		•
SetupItemCaptions(ReopenMenuIte m);   SetupItemCaptions(ReopenMenuIte m);	1.002	_
if ReopenItem = nil then exit;  // Add thew new filename to the beginning of the list and move other items down  for I := ReopenActionList1.ActionCount - 1 downto 0 do     if I = 0 then  TCustomAction(ReopenActionList1. Actions[I]).Caption := FileOpen1.Dialog.FileName     else  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  if necessary if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add; // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  SetupItemCaptions(ReopenMenuIte m);  SetupItemCaptions(ReopenMenuIte m);  SetupItemCaptions(ReopenMenuIte m);  SetupItemCaptions(ReopenMenuIte m);  SetupItemCaptions(PopenToolItem); end;  SetupItemCaptions(PopenToolItem);  end;  SetupItemCaptions(PopenToolItem);  end;  I: Integer; Found: = False;  // If the filename is already in the list then do not add it again for I := 0 to ReopenActionList1.ActionCount - 1 do  if CompareText(TCustomAction(ReopenActionList1.Actions[I]).Caption,  FileOpen1.Dialog.FileName		since they are used to open the the
// Add thew new filename to the beginning of the list and move other items down  for I := ReopenActionList1.ActionCount - 1 downto 0 do     if I = 0 then  TCustomAction(ReopenActionList1. Actions[I]).Caption := FileOpen1.Dialog.FileName else  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I - 1]).Caption;  // Add new items to the reopen item if necessary  if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  m);  SetupItemCaptions(OpenToolItem); end;  Var  I: Integer; Found := False;  // If the filename is already in the list then do not add it again  for I := 0 to ReopenActionList1.ActionCount - 1 do  if CompareText(TCustomAction(ReopenActionList1.Actions[I]).Caption, FileOpen1.Dialog.FileName) = 0 then  begin  Found := True;  begin  Found := True;  break; end; if not Found then  begin		S-tItCtiDMIt-
beginning of the list and move other items down  for I := ReopenActionList1.ActionCount - 1 downto 0 do  if I = 0 then  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I]).Caption:  TCustomAction(ReopenActionList1. Actions[I]).Caption:  TCustomAction(ReopenActionList1. Actions[I]).Caption:  if the filename is already in the list then do not add it again  for I := 0 to ReopenActionList1.ActionCount - 1 do  if CompareText(TCustomAction(ReopenActionList1.ActionList1.Actions[I]).Caption, FileOpen1.Dialog.FileName) = 0 then  begin  Found := True; break; end;  if not Found then  begin		
items down  for I := ReopenActionList1.ActionCount - 1 downto 0 do  if I = 0 then  TCustomAction(ReopenActionList1. Actions[I]).Caption := FileOpen1.Dialog.FileName else  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I - 1]).Caption;  // Add new items to the reopen item if necessary if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  SetupItemCaptions(OpenToolItem); end;  end;  procedure TForm1.FileOpen1Accept(Sender: TObject);  var  I: Integer; Found := False;  // If the filename is already in the list then do not add it again  for I := 0 to ReopenActionList1.ActionCount - 1 do  if CompareText(TCustomAction(ReopenActionList1.Actions[I]).Caption, FileOpen1.Dialog.FileName) = 0 then  begin  Found := True; break; end; if not Found then		m);
for I := ReopenActionList1.ActionCount - 1 downto 0 do  if I = 0 then  TCustomAction(ReopenActionList1. Actions[I]).Caption := FileOpen1.Dialog.FileName else  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I - 1]).Caption;  // Add new items to the reopen item if necessary if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  end;  procedure TForm1.FileOpen1Accept(Sender: TObject);  var  I: Integer; Found: Boolean; begin  Found := False;  // If the filename is already in the list then do not add it again for I := 0 to ReopenActionList1.ActionCount - 1 do ReopenActionList1.ActionCount - 1 do  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store		
ReopenActionList1.ActionCount - 1 downto 0 do  if I = 0 then  TCustomAction(ReopenActionList1. Actions[I]).Caption := FileOpen1.Dialog.FileName else  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I - 1]).Caption :=  TCustomAction(ReopenActionList1. Actions[I - 1]).Caption;  // Add new items to the reopen item if necessary if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  procedure TForm1.FileOpen1Accept(Sender: TObject);  var  I: Integer; Found: Boolean; begin  Found := False;  // If the filename is already in the list then do not add it again for I := 0 to ReopenActionList1.ActionCount - 1 do  if CompareText(TCustomAction(ReopenActionList1.Actions[I]).Caption, FileOpen1.Dialog.FileName) = 0 then  begin  Found := True; break; end; if not Found then		• • • • • • • • • • • • • • • • • • • •
downto 0 do  if I = 0 then  TCustomAction(ReopenActionList1. Actions[I]).Caption := FileOpen1.Dialog.FileName else  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I - 1]).Caption;  // Add new items to the reopen item if necessary  if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  procedure TForm1.FileOpen1Accept(Sender: TObject);  var  I: Integer; Found: Boolean; begin  Found := False;  // If the filename is already in the list then do not add it again for I := 0 to ReopenActionList1.ActionCount - 1 do  if CompareText(TCustomAction(ReopenActionList1.Actions[I]).Caption, FileOpen1.Dialog.FileName) = 0 then  begin  Found := True; break; end; if not Found then		end;
if I = 0 then  TCustomAction(ReopenActionList1. Actions[I]).Caption := FileOpen1.Dialog.FileName else  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I - 1]).Caption;  // Add new items to the reopen item if necessary if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  TForm1.FileOpen1Accept(Sender: TObject);  var  I: Integer; Found: Boolean; begin  Found := False;  // If the filename is already in the list then do not add it again for I := 0 to ReopenActionList1.ActionCount - 1 do  ReopenActionList1.ActionCount - 1 do  TForm1.FileOpen1Accept(Sender: TObject);  var  I: Integer; Found: Boolean;  begin  Found := False;  // If the filename is already in the list then do not add it again for I := 0 to ReopenActionList1.ActionCount - 1 do  TEOMIC: TOUSION:		
TCustomAction(ReopenActionList1. Actions[I]).Caption := FileOpen1.Dialog.FileName else  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I].Caption :=  TCustomAction(ReopenActionList1. Actions[I]-1]).Caption;  // Add new items to the reopen item if necessary if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  TObject);  var  I: Integer; Found: Boolean; begin  Found := False;  // If the filename is already in the list then do not add it again  for I := 0 to ReopenActionList1.ActionCount - 1 do  ReopenActionList1.ActionCount - 1 fileOpen1.Dialog.FileName) = 0 then begin  TObject);  var  I: Integer; Found: Boolean;  begin	downto 0 do	-
TCustomAction(ReopenActionList1. Actions[I]).Caption :=  FileOpen1.Dialog.FileName  else  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I - 1]).Caption:  TCustomAction(ReopenActionList1. Actions[I - 1]).Caption;  // Add new items to the reopen item if necessary  if ReopenItem.Items.Count <  ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store    Var	if I = 0 then	
Actions[I]).Caption := FileOpen1.Dialog.FileName else  TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I-1]).Caption;  // Add new items to the reopen item if necessary if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  I: Integer; Found: Boolean;  begin  Found:= False;  // If the filename is already in the list then do not add it again  for I := 0 to ReopenActionList1.ActionCount - 1 do  if CompareText(TCustomAction(ReopenActionList1.Actions[I]).Caption, FileOpen1.Dialog.FileName) = 0 then  begin  Found:= True; break; end; if not Found then begin		TObject);
Found: Boolean;  begin  Found := False;  // If the filename is already in the list then do not add it again  for I := 0 to  ReopenActionList1. Actions[I - 1]).Caption;  // Add new items to the reopen item if necessary  if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  Found: Boolean;  begin  Found: = False;  // If the filename is already in the list then do not add it again  for I := 0 to  ReopenActionList1.ActionCount - 1  do  if  CompareText(TCustomAction(ReopenActionList1.Actions[I]).Caption,  FileOpen1.Dialog.FileName) = 0  then  begin  Found: = True;  break;  end;  if not Found then  begin	TCustomAction(ReopenActionList1.	var
Found: Boolean;  begin  Found := False;  // If the filename is already in the list then do not add it again  for I := 0 to  ReopenActionList1. Actions[I - 1]).Caption;  // Add new items to the reopen item if necessary  if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  Found: Boolean;  begin  Found: = False;  // If the filename is already in the list then do not add it again  for I := 0 to  ReopenActionList1.ActionCount - 1  do  if  CompareText(TCustomAction(ReopenActionList1.Actions[I]).Caption,  FileOpen1.Dialog.FileName) = 0  then  begin  Found: = True;  break;  end;  if not Found then  begin	Actions[I]).Caption :=	I: Integer;
TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I - 1]).Caption;  // Add new items to the reopen item if necessary  if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  Found := False;  // If the filename is already in the list then do not add it again  for I := 0 to ReopenActionList1.ActionCount - 1 do  ReopenActionList1.ActionCount - 1 do  FileOpen1.Dialog.FileName) = 0 then  begin  Found := False;  // If the filename is already in the list then do not add it again  for I := 0 to ReopenActionList1.ActionCount - 1 do  FileOpen1.Dialog.FileName) = 0 then  begin  Found := False;	FileOpen1.Dialog.FileName	Found: Boolean;
TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I - 1]).Caption;  // Add new items to the reopen item if necessary  if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  Found := False;  // If the filename is already in the list then do not add it again  for I := 0 to ReopenActionList1.ActionCount - 1 do  ReopenActionList1.ActionCount-1 do  FileOpen1.Dialog.FileName) = 0 then  begin  Found := True;  break; end; if not Found then begin	else	begin
TCustomAction(ReopenActionList1. Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I - 1]).Caption;  // Add new items to the reopen item if necessary  if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  // If the filename is already in the list then do not add it again for I := 0 to ReopenActionList1.ActionCount - 1 do  ReopenActionList1.ActionCount - 1 do  ReopenActionList1.ActionS[I]).Caption, FileOpen1.Dialog.FileName) = 0 then  begin  Found := True; break; end; if not Found then begin		
Actions[I]).Caption :=  TCustomAction(ReopenActionList1. Actions[I - 1]).Caption;  // Add new items to the reopen item if necessary  if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  list then do not add it again  for I := 0 to ReopenActionList1.ActionCount - 1 do  if CompareText(TCustomAction(Reop enActionList1.Actions[I]).Caption, FileOpen1.Dialog.FileName) = 0 then  begin  Found := True; break; end; if not Found then begin	TCustomAction(ReopenActionList1.	·
TCustomAction(ReopenActionList1. Actions[I - 1]).Caption;  // Add new items to the reopen item if necessary  if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  for I := 0 to ReopenActionList1.ActionCount - 1 do  SeptemactionList1.ActionCount - 1 do  If CompareText(TCustomAction(Reop enActionList1.Actions[I]).Caption,  FileOpen1.Dialog.FileName) = 0 then  begin  for I := 0 to  ReopenActionList1.ActionCount - 1 do  If CompareText(TCustomAction(Reop enActionList1.Actions[I]).Caption,  FileOpen1.Dialog.FileName) = 0 then  begin  for I := 0 to  ReopenActionList1.ActionCount - 1 do  If CompareText(TCustomAction(Reop enActionList1.Actions[I]).Caption,  FileOpen1.Dialog.FileName) = 0 then  begin  for I := 0 to  ReopenActionList1.ActionCount - 1 do  If CompareText(TCustomAction(Reop enActionList1.Actions[I]).Caption,  FileOpen1.Dialog.FileName) = 0 then  begin  for I := 0 to  ReopenActionList1.ActionCount - 1 do  If CompareText(TCustomAction(Reop enActionList1.Actions[I]).Caption,  FileOpen1.Dialog.FileName) = 0 then  begin  for I := 0 to  ReopenActionList1.ActionCount - 1 do  If CompareText(TCustomAction(Reop enActionList1.Actions[I]).Caption,  FileOpen1.Dialog.FileName) = 0 then  begin	` -	•
TCustomAction(ReopenActionList1. Actions[I - 1]).Caption;  // Add new items to the reopen item if necessary  if ReopenItem.Items.Count < CompareText(TCustomAction(ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  ReopenActionList1.ActionCount if do  CompareText(TCustomAction(ReopenActionList1.Actions[I]).Caption,  FileOpen1.Dialog.FileName) = 0 then  begin  Found := True;  break;  end;  if not Found then  begin	L W Free C	
Actions[I - 1]).Caption;  // Add new items to the reopen item if necessary  if ReopenItem.Items.Count <	TCustomAction(ReonenActionList1	
// Add new items to the reopen item if necessary  if ReopenItem.Items.Count <	` =	_
if necessary  if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  CompareText(TCustomAction(Reop enActionList1.Actions[I]).Caption,  FileOpen1.Dialog.FileName) = 0 then  begin  Found := True;  break; end; if not Found then  begin		
if ReopenItem.Items.Count < ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  enActionList1.Actions[I]).Caption,  FileOpen1.Dialog.FileName) = 0 then  begin  Found := True;  break; end; if not Found then begin	-	
ReopenActionList1.ActionCount then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  FileOpen1.Dialog.FileName) = 0 then  begin  Found := True;  break;  end;  if not Found then  begin		` ` ` `
then  ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  then begin  Found := True; break; end; if not Found then begin	-	
ReopenItem.Items.Add;  // Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  begin  Found := True; break; end; if not Found then begin	=	1 9
// Set the item captions by appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  Found := True; break; end; if not Found then begin		
appending a number for use as the shortcut  // This change will cause them to be streamed which allows us to store  break; end; if not Found then begin		
shortcut  // This change will cause them to be streamed which allows us to store  end; if not Found then begin	<u> </u>	,
// This change will cause them to be streamed which allows us to store begin		,
streamed which allows us to store begin		· · · · · · · · · · · · · · · · · · ·
begin	_	
the // Update the Reopen menu		
	the	// Update the Reopen menu

Width = 499
Height = 285
Caption = 'ActionBands Reopen
Demo'
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = False
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
object ActionMainMenuBar1:
TActionMainMenuBar
Left = 0
Top = 0
Width = 491
Height = 25
UseSystemFont = False
ActionManager =
ActionManager1
Caption = 'ActionMainMenuBar1'
ColorMap.HighlightColor =
ColorMap.HighlightColor = 15660791
ColorMap.HighlightColor = 15660791 ColorMap.BtnSelectedColor =
ColorMap.HighlightColor = 15660791 ColorMap.BtnSelectedColor = clBtnFace
ColorMap.HighlightColor = 15660791 ColorMap.BtnSelectedColor = clBtnFace ColorMap.UnusedColor =
ColorMap.HighlightColor = 15660791  ColorMap.BtnSelectedColor = clBtnFace  ColorMap.UnusedColor = 15660791
ColorMap.HighlightColor = 15660791  ColorMap.BtnSelectedColor = clBtnFace  ColorMap.UnusedColor = 15660791  EdgeBorders = [ebBottom]
ColorMap.HighlightColor = 15660791  ColorMap.BtnSelectedColor = clBtnFace  ColorMap.UnusedColor = 15660791  EdgeBorders = [ebBottom] Font.Charset =
ColorMap.HighlightColor = 15660791  ColorMap.BtnSelectedColor = clBtnFace  ColorMap.UnusedColor = 15660791  EdgeBorders = [ebBottom]  Font.Charset = DEFAULT_CHARSET
ColorMap.HighlightColor = 15660791  ColorMap.BtnSelectedColor = clBtnFace  ColorMap.UnusedColor = 15660791  EdgeBorders = [ebBottom] Font.Charset =
ColorMap.HighlightColor = 15660791  ColorMap.BtnSelectedColor = clBtnFace  ColorMap.UnusedColor = 15660791  EdgeBorders = [ebBottom]  Font.Charset = DEFAULT_CHARSET
ColorMap.HighlightColor = 15660791  ColorMap.BtnSelectedColor = clBtnFace  ColorMap.UnusedColor = 15660791  EdgeBorders = [ebBottom] Font.Charset = DEFAULT_CHARSET Font.Color = clWindowText
ColorMap.HighlightColor = 15660791  ColorMap.BtnSelectedColor = clBtnFace  ColorMap.UnusedColor = 15660791  EdgeBorders = [ebBottom] Font.Charset = DEFAULT_CHARSET Font.Color = clWindowText Font.Height = -11
ColorMap.HighlightColor = 15660791  ColorMap.BtnSelectedColor = clBtnFace  ColorMap.UnusedColor = 15660791  EdgeBorders = [ebBottom]  Font.Charset = DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'Tahoma'
ColorMap.HighlightColor = 15660791  ColorMap.BtnSelectedColor = clBtnFace  ColorMap.UnusedColor = 15660791  EdgeBorders = [ebBottom]  Font.Charset = DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'Tahoma'  Font.Style = []  Spacing = 3 end
ColorMap.HighlightColor = 15660791  ColorMap.BtnSelectedColor = clBtnFace  ColorMap.UnusedColor = 15660791  EdgeBorders = [ebBottom]  Font.Charset = DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'Tahoma'  Font.Style = []  Spacing = 3 end
ColorMap.HighlightColor = 15660791  ColorMap.BtnSelectedColor = clBtnFace  ColorMap.UnusedColor = 15660791  EdgeBorders = [ebBottom]  Font.Charset = DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'Tahoma'  Font.Style = [] Spacing = 3
ColorMap.HighlightColor = 15660791  ColorMap.BtnSelectedColor = clBtnFace  ColorMap.UnusedColor = 15660791  EdgeBorders = [ebBottom]  Font.Charset = DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'Tahoma'  Font.Style = []  Spacing = 3 end object ActionToolBar1:
ColorMap.HighlightColor = 15660791  ColorMap.BtnSelectedColor = clBtnFace  ColorMap.UnusedColor = 15660791  EdgeBorders = [ebBottom]  Font.Charset = DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'Tahoma'  Font.Style = []  Spacing = 3 end object ActionToolBar1: TActionToolBar  Left = 0
ColorMap.HighlightColor = 15660791  ColorMap.BtnSelectedColor = clBtnFace  ColorMap.UnusedColor = 15660791  EdgeBorders = [ebBottom]  Font.Charset = DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'Tahoma'  Font.Style = []  Spacing = 3 end object ActionToolBar1: TActionToolBar
ColorMap.HighlightColor = 15660791  ColorMap.BtnSelectedColor = clBtnFace  ColorMap.UnusedColor = 15660791  EdgeBorders = [ebBottom]  Font.Charset = DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'Tahoma'  Font.Style = []  Spacing = 3 end object ActionToolBar1: TActionToolBar  Left = 0  Top = 25 Width = 491
ColorMap.HighlightColor = 15660791  ColorMap.BtnSelectedColor = clBtnFace  ColorMap.UnusedColor = 15660791  EdgeBorders = [ebBottom] Font.Charset = DEFAULT_CHARSET  Font.Color = clWindowText Font.Height = -11 Font.Name = 'Tahoma' Font.Style = [] Spacing = 3 end object ActionToolBar1: TActionToolBar  Left = 0  Top = 25 Width = 491 Height = 26
ColorMap.HighlightColor = 15660791  ColorMap.BtnSelectedColor = clBtnFace  ColorMap.UnusedColor = 15660791  EdgeBorders = [ebBottom]  Font.Charset = DEFAULT_CHARSET  Font.Color = clWindowText  Font.Height = -11  Font.Name = 'Tahoma'  Font.Style = []  Spacing = 3 end object ActionToolBar1: TActionToolBar  Left = 0  Top = 25 Width = 491

// filenames when the application is
shutdown
for I := 0 to
ReopenItem.Items.Count - 1 do
begin
ReopenItem.Items[I].Action :=
ReopenActionList1.Actions[I];
ReopenItem.Items[I].Caption :=
Format('&%d: %s', [I,
TCustomAction(ReopenActionList1. Actions[I]).Caption]);
end;
end;
_
procedure
TForm1.ReopenActionExecute(Send
er: TObject);
begin LGI : 4
// Set the reopened filename into
the FileOpen action and call
OnAccept to open the file normally
FileOpen1.Dialog.FileName :=
(Sender as TCustomAction).Caption;
// Execute the action's OnAccept
logic
FileOpen1.OnAccept(nil);
end;
ciiu,
end.
Ciru.
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

<b>ObjectS</b>	
الهلف النصي لبرناهج	
(MRU)	
إعداد	
علاء الدين اللباد	
واتف ۷۳۷۱۹۹۱	
MRU	
object Form1: TForm1	
Left = 187	
Top = 200	

object Action8: TAction
Caption = 'Action8'
OnExecute =
ReopenActionExecute
end
end
object ActionManager1:
TActionManager
FileName = 'c:\code\actions\action
bars\reopen\settings.dat'
ActionBars = <
item
Items = <
item
ChangesAllowed =
[caModify]
Items.HideUnused = False
Items = <
item
Action = FileOpen1
ImageIndex = 0
ShortCut = 16463
end
item
ChangesAllowed = [caModify]
Caption = '&Reopen'
end
item
Caption = '-'
end
item
Action = FileExit1
ImageIndex = 19
end>
Caption = '&File'
end
item
Items = <
item
Action = EditCut1
ImageIndex = 1
ShortCut = 16472
end
item
Action = EditCopy1
ImageIndex = 2

Caption = 'ActionToolBar1'
ColorMap.HighlightColor =
15660791
ColorMap.BtnSelectedColor =
clBtnFace
ColorMap.UnusedColor =
15660791
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
Spacing = 0
end
object RichEdit1: TRichEdit
Left = 9
Top = 65
Width = 371
Height = 182
Lines.Strings = (
'RichEdit1')
TabOrder = 2
end
object ReopenActionList1:
TActionList
Left = 91
Top = 94
object Action4: TAction
Caption = 'Action4'
OnExecute =
ReopenActionExecute
end
object Action5: TAction
Caption = 'Action5'
OnExecute =
ReopenActionExecute
end
object Action6: TAction
Caption = 'Action6' OnExecute =
ReopenActionExecute
end
object Action7: TAction
Caption = 'Action7'
OnExecute =
ReopenActionExecute

ShortCut = 16450 end item Action = RichEditItalic1 ImageIndex = 7 ShortCut = 16457 end item Action = RichEditUnderline1 ImageIndex = 8 ShortCut = 16469 end item Action = RichEditStrikeOut1 ImageIndex = 9 end item Action = RichEditBullets1 Caption = 'Bu&llets' ImageIndex = 10 end item Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item  Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item  Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item  Action = CustomizeActionBars1 end>	I I I (
end item  Action = RichEditItalic1  ImageIndex = 7 ShortCut = 16457 end item  Action = RichEditUnderline1 ImageIndex = 8 ShortCut = 16469 end item  Action = RichEditStrikeOut1 ImageIndex = 9 end item  Action = RichEditBullets1 Caption = 'Bu&llets' ImageIndex = 10 end item  Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item  Action = RichEditAlignRight1 ImageIndex = 12 end item  Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item  Items = < item  Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item  Items = < item  Action = CustomizeActionBars1	ImageIndex = 6
item  Action = RichEditItalic1  ImageIndex = 7  ShortCut = 16457  end item  Action = RichEditUnderline1  ImageIndex = 8  ShortCut = 16469  end item  Action = RichEditStrikeOut1  ImageIndex = 9  end item  Action = RichEditBullets1  Caption = 'Bu&llets'  ImageIndex = 10  end item  Action = RichEditAlignLeft1  Caption = '&Align Left'  ImageIndex = 11  end item  Action = RichEditAlignRight1  ImageIndex = 12  end item  Action = RichEditAlignCenter1  ImageIndex = 13  end>  Caption = 'F&ormat' end item  Items = < item  Action = CustomizeActionBars1	
Action = RichEditItalic1 ImageIndex = 7 ShortCut = 16457 end item Action = RichEditUnderline1 ImageIndex = 8 ShortCut = 16469 end item Action = RichEditStrikeOut1 ImageIndex = 9 end item Action = RichEditBullets1 Caption = 'Bu&llets' ImageIndex = 10 end item Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Action = CustomizeActionBars1	V == 47
ImageIndex = 7 ShortCut = 16457 end item Action = RichEditUnderline1 ImageIndex = 8 ShortCut = 16469 end item Action = RichEditStrikeOut1 ImageIndex = 9 end item Action = RichEditBullets1 Caption = 'Bu&llets' ImageIndex = 10 end item Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	100111
ShortCut = 16457 end item Action = RichEditUnderline1 ImageIndex = 8 ShortCut = 16469 end item Action = RichEditStrikeOut1 ImageIndex = 9 end item Action = RichEditBullets1 Caption = 'Bu&llets' ImageIndex = 10 end item Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
end item Action = RichEditUnderline1 ImageIndex = 8 ShortCut = 16469 end item Action = RichEditStrikeOut1 ImageIndex = 9 end item Action = RichEditBullets1 Caption = 'Bu&llets' ImageIndex = 10 end item Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
item Action = RichEditUnderline1 ImageIndex = 8 ShortCut = 16469 end item Action = RichEditStrikeOut1 ImageIndex = 9 end item Action = RichEditBullets1 Caption = 'Bu&llets' ImageIndex = 10 end item Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Items = < item CustomizeActionBars1	<b>ShortCut = 16457</b>
Action = RichEditUnderline1 ImageIndex = 8 ShortCut = 16469 end item Action = RichEditStrikeOut1 ImageIndex = 9 end item Action = RichEditBullets1 Caption = 'Bu&llets' ImageIndex = 10 end item Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Items = < item CustomizeActionBars1	end
RichEditUnderline1 ImageIndex = 8 ShortCut = 16469 end item Action = RichEditStrikeOut1 ImageIndex = 9 end item Action = RichEditBullets1 Caption = 'Bu&llets' ImageIndex = 10 end item Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
ImageIndex = 8 ShortCut = 16469 end item Action = RichEditStrikeOut1 ImageIndex = 9 end item Action = RichEditBullets1 Caption = 'Bu&llets' ImageIndex = 10 end item Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	11001011
ShortCut = 16469 end item Action = RichEditStrikeOut1 ImageIndex = 9 end item Action = RichEditBullets1 Caption = 'Bu&llets' ImageIndex = 10 end item Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	RichEditUnderline1
end item Action = RichEditStrikeOut1 ImageIndex = 9 end item Action = RichEditBullets1 Caption = 'Bu&llets' ImageIndex = 10 end item Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
item Action = RichEditStrikeOut1 ImageIndex = 9 end item Action = RichEditBullets1 Caption = 'Bu&llets' ImageIndex = 10 end item Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	ShortCut = 16469
Action = RichEditStrikeOut1  ImageIndex = 9 end item Action = RichEditBullets1 Caption = 'Bu&llets' ImageIndex = 10 end item Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	end
RichEditStrikeOut1 ImageIndex = 9 end item Action = RichEditBullets1 Caption = 'Bu&llets' ImageIndex = 10 end item Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	item
ImageIndex = 9 end item Action = RichEditBullets1 Caption = 'Bu&llets' ImageIndex = 10 end item Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	Action =
end item  Action = RichEditBullets1 Caption = 'Bu&llets' ImageIndex = 10 end item  Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	RichEditStrikeOut1
item    Action = RichEditBullets1    Caption = 'Bu&llets'    ImageIndex = 10    end    item    Action =    RichEditAlignLeft1    Caption = '&Align Left'    ImageIndex = 11    end    item    Action =    RichEditAlignRight1    ImageIndex = 12    end    item    Action =    RichEditAlignCenter1    ImageIndex = 13    end>    Caption = 'F&ormat'    end    item    Items = <    item    Action =    CustomizeActionBars1	ImageIndex = 9
Action = RichEditBullets1 Caption = 'Bu&llets' ImageIndex = 10 end item Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	Ü
Caption = 'Bu&llets' ImageIndex = 10 end item Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	item
Caption = 'Bu&llets' ImageIndex = 10 end item Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	Action = RichEditBullets1
ImageIndex = 10 end item  Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
end item  Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
item    Action = RichEditAlignLeft1    Caption = '&Align Left'    ImageIndex = 11    end    item    Action = RichEditAlignRight1    ImageIndex = 12    end    item    Action = RichEditAlignCenter1    ImageIndex = 13    end>    Caption = 'F&ormat'    end    item    Items = <    item    Action = CustomizeActionBars1	
Action = RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
RichEditAlignLeft1 Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
Caption = '&Align Left' ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
ImageIndex = 11 end item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
end item  Action = RichEditAlignRight1  ImageIndex = 12 end item  Action = RichEditAlignCenter1  ImageIndex = 13 end> Caption = 'F&ormat' end item  Items = < item  Action = CustomizeActionBars1	
item Action = RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
Action = RichEditAlignRight1  ImageIndex = 12 end item Action = RichEditAlignCenter1  ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
RichEditAlignRight1 ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
ImageIndex = 12 end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
end item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
item Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
Action = RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
RichEditAlignCenter1 ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
ImageIndex = 13 end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
end> Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
Caption = 'F&ormat' end item Items = < item Action = CustomizeActionBars1	
end item Items = < item Action = CustomizeActionBars1	
item  Items = < item  Action =  CustomizeActionBars1	
Items = < item Action = CustomizeActionBars1	
item Action = CustomizeActionBars1	_
Action = CustomizeActionBars1	_
CustomizeActionBars1	
end>	
	end>

<b>ShortCut = 16451</b>
end
item
Action = EditPaste1
ImageIndex = 3
ShortCut = 16470
end
item
Action = EditSelectAll1
ShortCut = 16449
end
item
Action = EditUndo1
ImageIndex = 4
ShortCut = 16474
end
item
Action = EditDelete1
ImageIndex = 5 ShortCut = 46
end>
Caption = '&Edit'
end
item
Items = <
item
Action = SearchFind1
ImageIndex = 14
ShortCut = 16454
end
item
Action = SearchFindNext1
ImageIndex = 15
ShortCut = 114
end
item
Action = SearchReplace1
ImageIndex = 16
end
item
Action = SearchFindFirst1
end>
Caption = '&Search'
end
item
Items = <
item
Action = RichEditBold1
ACHOH - KICHEURDORT

ShortCut = 16463
OnAccept = FileOpen1Accept
end
object EditCut1: TEditCut
Category = 'Edit'
Caption = 'Cu&t'
Hint = 'Cut Cuts the selection
and puts it on the Clipboard'
ImageIndex = 1
ShortCut = 16472
end
object EditCopy1: TEditCopy
Category = 'Edit'
Caption = '&Copy'
Hint = 'Copy Copies the selection
and puts it on the Clipboard'
ImageIndex = 2
ShortCut = 16451
end
object EditPaste1: TEditPaste
Category = 'Edit'
Caption = '&Paste'
Hint = 'Paste Inserts Clipboard
contents'
ImageIndex = 3
ShortCut = 16470
end
object EditSelectAll1:
TEditSelectAll
Category = 'Edit'
Caption = 'Select &All'
Hint = 'Select All Selects the
entire document'
ShortCut = 16449
end
object EditUndo1: TEditUndo
Category = 'Edit'
Caption = '&Undo'
Hint = 'Undo Reverts the last
action'
ImageIndex = 4
ShortCut = 16474
end
object EditDelete1: TEditDelete
Category = 'Edit'
Caption = '&Delete'
Hint = 'Delete Erases the
selection'

Caption = '&Tools'
end>
ActionBar =
ActionMainMenuBar1
end
item
Items.HideUnused = False
Items = <
item
Action = FileOpen1
ImageIndex = 0
ShortCut = 16463
end>
ActionBar = ActionToolBar1
end>
LinkedActionLists = <
item
ActionList = ReopenActionList1
Caption = 'Reopenlist'
end>
Images = ImageList1
Left = 259
Top = 101
StyleName = 'XP Style'
object Action1: TAction
Category = 'Reopen'
Caption = 'Action1'
OnExecute =
ReopenActionExecute
end
object Action2: TAction
Category = 'Reopen'
Caption = 'Action2'
OnExecute =
ReopenActionExecute
end
object Action3: TAction
Category = 'Reopen'
Caption = 'Action3'
OnExecute =
ReopenActionExecute
end
object FileOpen1: TFileOpen
Category = 'File'
Caption = '&Open'
Hint = 'Open Opens an existing
file'
ImageIndex = 0

TRichEditAlignLeft
Category = 'Format'
AutoCheck = True
Caption = 'Align &Left'
Hint = 'Align Left Aligns text at
the left indent'
ImageIndex = 11
end
object RichEditAlignRight1:
TRichEditAlignRight
Category = 'Format'
AutoCheck = True
Caption = 'Align & Right'
Hint = 'Align Right Aligns text at
the right indent'
ImageIndex = 12
end
object RichEditAlignCenter1:
TRichEditAlignCenter
Category = 'Format'
AutoCheck = True
Caption = '&Center'
Hint = 'Center Centers text
between margins'
ImageIndex = 13
end
object SearchFind1: TSearchFind
Category = 'Search'
Caption = '&Find'
Hint = 'Find Finds the specified
text'
ImageIndex = 14
<b>ShortCut = 16454</b>
end
object SearchFindNext1:
TSearchFindNext
Category = 'Search'
Caption = 'Find &Next'
Hint = 'Find Next Repeats the
last find'
ImageIndex = 15
SearchFind = SearchFind1
ShortCut = 114
end
object SearchReplace1:
TSearchReplace
Category = 'Search'
Caption = '&Replace'

ImageIndex = 5
ShortCut = 46
end
object RichEditBold1:
TRichEditBold
Category = 'Format'
AutoCheck = True
Caption = '&Bold'
Hint = 'Bold'
ImageIndex = 6
ShortCut = 16450
end
object RichEditItalic1:
TRichEditItalic
Category = 'Format'
AutoCheck = True
Caption = '&Italic'
Hint = 'Italic'
ImageIndex = 7
ShortCut = 16457
end
object RichEditUnderline1: TRichEditUnderline
Category = 'Format' AutoCheck = True
Caption = '&Underline'
Hint = 'Underline'
ImageIndex = 8
ShortCut = 16469
end
object RichEditStrikeOut1:
TRichEditStrikeOut
Category = 'Format'
AutoCheck = True
Caption = '&Strikeout'
Hint = 'Strikeout'
ImageIndex = 9
end
object RichEditBullets1:
TRichEditBullets
Category = 'Format'
AutoCheck = True
Caption = '&Bullets'
Hint = 'Bullets Inserts a bullet on
the current line'
ImageIndex = 10
end
object RichEditAlignLeft1:
object Mennant/Mighnetti.

end object FileExit1: TFileExit Category = 'File' Caption = 'E&xit' Hint = 'Exit Quits the application' ImageIndex = 19 end object Action9: TAction Category = 'File' Caption = 'Action9' end object CustomizeActionBars1: TCustomizeActionBars Category = 'Tools' Caption = '&Customize' CustomizeDlg.StayOnTop = False end end object ImageList1: TImageList Left = 178 Top = 147 Bitmap = {  494C010114001900040010001000FF FFFFFFFFFFFFFFFFFFFFFFF	
Category = 'File' Caption = 'E&xit' Hint = 'Exit Quits the application' ImageIndex = 19 end object Action9: TAction Category = 'File' Caption = 'Action9' end object CustomizeActionBars1: TCustomizeActionBars Category = 'Tools' Caption = '&Customize' CustomizeDlg.StayOnTop = False end end object ImageList1: TImageList Left = 178 Top = 147 Bitmap = {  494C010114001900040010001000FF FFFFFFFFFFFFFFFFFFFFFFF	
Caption = 'E&xit'  Hint = 'Exit Quits the application'  ImageIndex = 19 end object Action9: TAction Category = 'File' Caption = 'Action9' end object CustomizeActionBars1: TCustomizeActionBars Category = 'Tools' Caption = '&Customize' CustomizeDlg.StayOnTop = False end end object ImageList1: TImageList Left = 178 Top = 147 Bitmap = {  494C010114001900040010001000FF FFFFFFFFFFFFFFFFFFFFFFF	
Hint = 'Exit Quits the application'  ImageIndex = 19 end object Action9: TAction Category = 'File' Caption = 'Action9' end object CustomizeActionBars1: TCustomizeActionBars Category = 'Tools' Caption = '&Customize' CustomizeDlg.StayOnTop = False end object ImageList1: TImageList Left = 178 Top = 147 Bitmap = {  494C010114001900040010001000FF FFFFFFFFFFFFFFFFFFFFFFF	
application' ImageIndex = 19 end object Action9: TAction Category = 'File' Caption = 'Action9' end object CustomizeActionBars1: TCustomizeActionBars Category = 'Tools' Caption = '&Customize' CustomizeDlg.StayOnTop = False end end object ImageList1: TImageList Left = 178 Top = 147 Bitmap = {  494C010114001900040010001000FF FFFFFFFFFFFFFFFFFFFFFFF	Caption = 'E&xit'
ImageIndex = 19 end object Action9: TAction Category = 'File' Caption = 'Action9' end object CustomizeActionBars1: TCustomizeActionBars Category = 'Tools' Caption = '&Customize' CustomizeDlg.StayOnTop = False end end object ImageList1: TImageList Left = 178 Top = 147 Bitmap = {  494C010114001900040010001000FF FFFFFFFFFFFFFFFFFFFFFFF	Hint = 'Exit Quits the
end object Action9: TAction Category = 'File' Caption = 'Action9' end object CustomizeActionBars1: TCustomizeActionBars Category = 'Tools' Caption = '&Customize' CustomizeDlg.StayOnTop = False end object ImageList1: TImageList Left = 178 Top = 147 Bitmap = {  494C010114001900040010001000FF FFFFFFFFFFFFFFFFFFFFFFF	application'
object Action9: TAction Category = 'File' Caption = 'Action9' end object CustomizeActionBars1: TCustomizeActionBars Category = 'Tools' Caption = '&Customize' CustomizeDlg.StayOnTop = False end object ImageList1: TImageList Left = 178 Top = 147 Bitmap = {  494C010114001900040010001000FF FFFFFFFFFFFFFFFFFFFFFFF	ImageIndex = 19
Category = 'File' Caption = 'Action9' end object CustomizeActionBars1: TCustomizeActionBars Category = 'Tools' Caption = '&Customize' CustomizeDlg.StayOnTop = False end end object ImageList1: TImageList Left = 178 Top = 147 Bitmap = {  494C010114001900040010001000FF FFFFFFFFFFFFFFFFFFFFFFF	end
Category = 'File' Caption = 'Action9' end object CustomizeActionBars1: TCustomizeActionBars Category = 'Tools' Caption = '&Customize' CustomizeDlg.StayOnTop = False end end object ImageList1: TImageList Left = 178 Top = 147 Bitmap = {  494C010114001900040010001000FF FFFFFFFFFFFFFFFFFFFFFFF	object Action9: TAction
Caption = 'Action9' end object CustomizeActionBars1: TCustomizeActionBars Category = 'Tools' Caption = '&Customize' CustomizeDlg.StayOnTop = False end end object ImageList1: TImageList Left = 178 Top = 147 Bitmap = {  494C010114001900040010001000FF FFFFFFFFFFFFFFFFFFFFFFF	Category = 'File'
end object CustomizeActionBars1: TCustomizeActionBars Category = 'Tools' Caption = '&Customize' CustomizeDlg.StayOnTop = False end end object ImageList1: TImageList Left = 178 Top = 147 Bitmap = {  494C010114001900040010001000FF FFFFFFFFFFFFFFFFFFFFFFF	Caption = 'Action9'
TCustomizeActionBars Category = 'Tools' Caption = '&Customize' CustomizeDlg.StayOnTop = False end end object ImageList1: TImageList Left = 178 Top = 147 Bitmap = {  494C010114001900040010001000FF FFFFFFFFFFFFFFFFFFFFFFF	
TCustomizeActionBars Category = 'Tools' Caption = '&Customize' CustomizeDlg.StayOnTop = False end end object ImageList1: TImageList Left = 178 Top = 147 Bitmap = {  494C010114001900040010001000FF FFFFFFFFFFFFFFFFFFFFFFF	object CustomizeActionBars1:
Caption = '&Customize' CustomizeDlg.StayOnTop = False end end object ImageList1: TImageList Left = 178 Top = 147 Bitmap = {  494C010114001900040010001000FF FFFFFFFFFFFFFFFFFFFFFFF	
Caption = '&Customize' CustomizeDlg.StayOnTop = False end end object ImageList1: TImageList Left = 178 Top = 147 Bitmap = {  494C010114001900040010001000FF FFFFFFFFFFFFFFFFFFFFFFF	Category = 'Tools'
False	
end end object ImageList1: TImageList Left = 178 Top = 147 Bitmap = {  494C010114001900040010001000FF FFFFFFFFFFFFFFFFFFFFFFF	CustomizeDlg.StayOnTop =
end object ImageList1: TImageList Left = 178 Top = 147 Bitmap = {  494C010114001900040010001000FF FFFFFFFFF10FFFFFFFFFFFF	False
object ImageList1: TImageList  Left = 178  Top = 147  Bitmap = {  494C010114001900040010001000FF FFFFFFFFF10FFFFFFFFFFFF	end
Left = 178	end
Top = 147 Bitmap = {  494C010114001900040010001000FF FFFFFFFFFFFFFFFFFFFFFFF	object ImageList1: TImageList
Bitmap = {  494C010114001900040010001000FF FFFFFFFFF10FFFFFFFFFFFF	Left = 178
494C010114001900040010001000FF FFFFFFFF10FFFFFFFFFFFFF	Top = 147
494C010114001900040010001000FF FFFFFFFF10FFFFFFFFFFFFF	Bitmap = {
FFFFFFFF10FFFFFFFFFFFFFFFFFFFFFFFFFFFF	
424D3600 end	494C010114001900040010001000FF
end	FFFFFFFF10FFFFFFFFFFFFFFFF
	424D3600
end	end
	end

Hint = 'Replace Replaces specific
text with different text'
ImageIndex = 16
end
object SearchFindFirst1:
TSearchFindFirst
Category = 'Search'
Caption = 'F&ind First'
Hint = 'Find First Finds the first
occurance of specified text'
end
object FileSaveAs1: TFileSaveAs
Category = 'File'
Caption = 'Save &As'
Hint = 'Save As Saves the active
file with a new name'
ImageIndex = 18
end
object FilePrintSetup1:
TFilePrintSetup
Category = 'File'
Caption = 'Print Set&up'
Hint = 'Print Setup'
end
object FileRun1: TFileRun
Category = 'File'
Browse = False
BrowseDlg.Title = 'Run'
Caption = '&Run'
Hint = 'Run Runs an application'
Operation = 'open'
ShowCmd = scShowNormal

ObjectS	
الملف النصي لبرناهج	
(ActionBands)	
إعداد	

object ToolActionBar1:
TActionToolBar
Left = 0
Top = 24
Width = 557
Height = 28
ActionManager =
ActionManager1
Caption = 'ToolActionBar1'
ColorMap.HighlightColor =
clBtnHighlight
ColorMap.UnusedColor =
15988985
ColorMap.SelectedColor =
clHighlight
EdgeBorders = [ebLeft, ebTop,
ebRight, ebBottom
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
Spacing = 0
end
object ToolActionBar2:
TActionToolBar
Left = 0
Top = 52
Width = 557
Height = 28
ActionManager =
ActionManager1
Caption = 'ToolActionBar2'
ColorMap.HighlightColor =
ColorMap.HighlightColor = clBtnHighlight
ColorMap.HighlightColor =
ColorMap.HighlightColor = clBtnHighlight ColorMap.UnusedColor =
ColorMap.HighlightColor = clBtnHighlight ColorMap.UnusedColor = 15988985
ColorMap.HighlightColor = clBtnHighlight ColorMap.UnusedColor = 15988985 ColorMap.SelectedColor = clHighlight
ColorMap.HighlightColor = clBtnHighlight ColorMap.UnusedColor = 15988985 ColorMap.SelectedColor = clHighlight EdgeBorders = [ebLeft, ebTop,
ColorMap.HighlightColor = clBtnHighlight ColorMap.UnusedColor = 15988985 ColorMap.SelectedColor = clHighlight
ColorMap.HighlightColor = clBtnHighlight  ColorMap.UnusedColor = 15988985  ColorMap.SelectedColor = clHighlight  EdgeBorders = [ebLeft, ebTop, ebRight, ebBottom]
ColorMap.HighlightColor = clBtnHighlight  ColorMap.UnusedColor = 15988985  ColorMap.SelectedColor = clHighlight  EdgeBorders = [ebLeft, ebTop, ebRight, ebBottom]  Font.Charset = DEFAULT_CHARSET
ColorMap.HighlightColor = clBtnHighlight  ColorMap.UnusedColor = 15988985  ColorMap.SelectedColor = clHighlight  EdgeBorders = [ebLeft, ebTop, ebRight, ebBottom]  Font.Charset = DEFAULT_CHARSET  Font.Color = clWindowText
ColorMap.HighlightColor = clBtnHighlight  ColorMap.UnusedColor = 15988985  ColorMap.SelectedColor = clHighlight  EdgeBorders = [ebLeft, ebTop, ebRight, ebBottom]  Font.Charset = DEFAULT_CHARSET

علاء الدين اللباد
8اتة. ۷۳٬۹۷۵۶۹۰
ActionBands
object Form1: TForm1
Left = 202
Top = 114
Width = 565
Height = 368
Caption = 'ActionBars Demo'
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = False
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
object ActionMainMenuBar1:
TActionMainMenuBar
Left = 0
Top = 0
Width = 557
Height = 24
UseSystemFont = False
ActionManager =
ActionManager1
Caption = 'ActionMainMenuBar1'
ColorMap.HighlightColor =
clBtnHighlight
ColorMap.UnusedColor =
15988985  ColorMan SalastadColor =
ColorMap.SelectedColor = clHighlight
EdgeBorders = [ebBottom] EdgeOuter = esNone
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'Tahoma'
Font.Style = []
Spacing = 4
end
CHU

object ActionManager1:
TActionManager (
ActionBars.SessionCount = 6
ActionBars = <
item
Items = <
item
Items = <
item
Action = FileOpen1
ImageIndex = 30
ShortCut = 16463
end
item
Action = FileSaveAs1
ImageIndex = 8
end
item
Action = FileRun1
UsageCount = -1
end
item
Caption = '-'
end
item
Action = FileExit1
ImageIndex = 43
UsageCount = -1
end>
Caption = '&File'
end
item
Items = <
item
Action = EditCut1
ImageIndex = 0
ShortCut = 16472
end
item
Action = EditCopy1
ImageIndex = 1
ShortCut = 16451
end
item
Action = EditPaste1
ImageIndex = 2
ShortCut = 16470
end

Font.Style = []
ParentFont = False
Spacing = 0
end
object ToolActionBar3:
TActionToolBar
Left = 0
Top = 80
Width = 557
Height = 28
ActionManager =
ActionManager1
Caption = 'ToolActionBar3'
ColorMap.HighlightColor =
clBtnHighlight
ColorMap.UnusedColor =
15988985
ColorMap.SelectedColor =
clHighlight
EdgeBorders = [ebLeft, ebTop,
ebRight, ebBottom
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
Spacing = 0
end
object RichEdit1: TRichEdit
Left = 0
Top = 108
Width = 557
Height = 207
Align = alClient
BevelKind = bkTile
BorderStyle = bsNone
TabOrder = 4
end
object StatusBar1: TStatusBar
Left = 0
Top = 315
Width = 557
Height = 19
AutoHint = True
Panels = <>
end
CHU

and d
end
item
Action =
RichEditUnderline1
ImageIndex = 28
ShortCut = 16469
end
item
Action =
RichEditStrikeOut1
end
item
Action = RichEditBullets1
Caption = 'Bull&ets'
ImageIndex = 38
end
item
Action =
RichEditAlignLeft1
ImageIndex = 35
end
item
Action =
RichEditAlignRight1
ImageIndex = 36
end
item
Action =
RichEditAlignCenter1
ImageIndex = 37
end>
Caption = 'F&ormat'
end
item
Items = <
item
Action =
CustomizeActionBars1
UsageCount = -1 end>
Caption = '&Tools'
end
item
Items = <
item
Action = StdStyleActn
Caption = '&Standard
Style'

item
Action = EditSelectAll1
<b>ShortCut = 16449</b>
end
item
Action = EditUndo1
ImageIndex = 3
<b>ShortCut = 16474</b>
end
item
Action = EditDelete1
ImageIndex = 5
ShortCut = 46
end>
Caption = '&Edit'
end
item
Items = <
item
Action = SearchFind1
ImageIndex = 34
ShortCut = 16454
end
item
Action = SearchFindNext1
ImageIndex = 33
ShortCut = 114
end
item
Action = SearchReplace1
ImageIndex = 32
end
item
Action = SearchFindFirst1
end>
Caption = '&Search'
end
item
Items = <
item
Action = RichEditBold1
ImageIndex = 31
ShortCut = 16450
end
item
Action = RichEditItalic1
ImageIndex = 29
ShortCut = 16457
2001000

Y Y 25
ImageIndex = 35
end
item
Action =
RichEditAlignRight1
ImageIndex = 36
end
item
Action =
RichEditAlignCenter1
ImageIndex = 37
end>
ActionBar = ToolActionBar1
end
item
Items = <
item
Action = SearchFind1
ImageIndex = 34
ShortCut = 16454
end
item
Action = SearchFindNext1
ImageIndex = 33
ShortCut = 114
end
item
Action = SearchReplace1
ImageIndex = 32
end
item
Caption = '-'
LastSession = 6
end
item
Action = SearchFindFirst1
end>
ActionBar = ToolActionBar2
AutoSize = False end
item
Items = <
item
Action = EditCut1
ImageIndex = 0
ShortCut = 16472
end
item

LastSession = 6
end
item
Action = XPStyleActn
Caption = '&XP Style'
LastSession = 6
end
item
Caption = '-'
end
item
Action = ShadowActn
Caption = '&Menu
Shadows'
LastSession = 6
end>
Caption = 'Sty≤'
end>
ActionBar =
ActionMainMenuBar1
end
item
Items = <
item
Action = RichEditBold1
ImageIndex = 31
ShortCut = 16450
end
item
Action = RichEditItalic1
ImageIndex = 29
ShortCut = 16457
end
item
Action = RichEditUnderline1
ImageIndex = 28
<b>ShortCut = 16469</b>
end
item
Action = RichEditStrikeOut1
end
item
Action = RichEditBullets1
Caption = 'Bull&ets'
ImageIndex = 38
end
item
Action = RichEditAlignLeft1
rection inclinated inglification

ShortCut = 16451
end
object EditPaste1: TEditPaste
Category = 'Edit'
Caption = '&Paste'
Enabled = False
Hint = 'Paste Inserts Clipboard
contents'
ImageIndex = 2
<b>ShortCut = 16470</b>
end
object EditSelectAll1:
TEditSelectAll
Category = 'Edit'
Caption = 'Select &All'
Enabled = False
Hint = 'Select All Selects the
entire document'
ShortCut = 16449
end
object EditUndo1: TEditUndo
Category = 'Edit'
Caption = '&Undo'
Enabled = False
Hint = 'Undo Reverts the last
action'
ImageIndex = 3
ShortCut = 16474
end
object EditDelete1: TEditDelete
Category = 'Edit'
Category - Edit  Caption = '&Delete'
Enabled = False
Hint = 'Delete Erases the selection'
ImageIndex = 5
ShortCut = 46
end
object RichEditBold1:
TRichEditBold
Category = 'Format'
AutoCheck = True
Caption = '&Bold'
Enabled = False
Hint = 'Bold'
ImageIndex = 31
ShortCut = 16450
end

Action = EditCopy1
ImageIndex = 1
ShortCut = 16451
end
item
Action = EditPaste1
ImageIndex = 2
ShortCut = 16470
end
item
Action = EditSelectAll1
ShortCut = 16449
end
item
Action = EditUndo1
ImageIndex = 3
ShortCut = 16474
end
item
Action = EditDelete1
ImageIndex = 5
ShortCut = 46
end>
ActionBar = ToolActionBar3
AutoSize = False
end>
Images = ImageList1
OnStateChange =
ActionManager1StateChange
Left = 125
Top = 138
StyleName = 'Standard'
object EditCut1: TEditCut
Category = 'Edit'
Caption = 'Cu&t'
Enabled = False
Hint = 'Cut Cuts the selection
and puts it on the Clipboard'
ImageIndex = 0
ShortCut = 16472
end
object EditCopy1: TEditCopy
Category = 'Edit'
Caption = '&Copy'
Enabled = False
Hint = 'Copy Copies the selection
and puts it on the Clipboard'
ImageIndex = 1

object RichEditAlignRight1:
TRichEditAlignRight
Category = 'Format'
AutoCheck = True
Caption = 'Align & Right'
Enabled = False
Hint = 'Align Right Aligns text at
the right indent'
ImageIndex = 36
end
object RichEditAlignCenter1:
TRichEditAlignCenter
Category = 'Format'
AutoCheck = True
Caption = '&Center'
Enabled = False
Hint = 'Center Centers text
between margins'
ImageIndex = 37
end
object FileOpen1: TFileOpen
Category = 'File'
Caption = '&Open'
Hint = 'Open Opens an existing
file'
ImageIndex = 30
ShortCut = 16463
OnAccept = FileOpen1Accept
end
object FileSaveAs1: TFileSaveAs
Category = 'File'
Caption = 'Save &As'
Hint = 'Save As Saves the active
file with a new name'
ImageIndex = 8
OnAccept = FileSaveAs1Accept
end
object FileRun1: TFileRun
Category = 'File'
Browse = False
BrowseDlg.Title = 'Run'
Caption = '&Run'
Hint = 'Run Runs an applicatoin'
Operation = 'open'
ShowCmd = scShowNormal
end
object FileExit1: TFileExit
Category = 'File'
Category Inc

object RichEditItalic1:
TRichEditItalic
Category = 'Format'
AutoCheck = True
Caption = '&Italic'
Enabled = False
Hint = 'Italic'
ImageIndex = 29
ShortCut = 16457
end
object RichEditUnderline1:
TRichEditUnderline
Category = 'Format'
AutoCheck = True
Caption = '&Underline'
Enabled = False
Hint = 'Underline'
ImageIndex = 28
ShortCut = 16469
end
object RichEditStrikeOut1:
TRichEditStrikeOut
Category = 'Format'
AutoCheck = True
Caption = '&Strikeout'
Enabled = False
Hint = 'Strikeout'
end
object RichEditBullets1:
TRichEditBullets
Category = 'Format'
AutoCheck = True
Caption = '&Bullets'
Enabled = False
Hint = 'Bullets Inserts a bullet on
the current line'
ImageIndex = 38
end
object RichEditAlignLeft1:
TRichEditAlignLeft
Category = 'Format'
AutoCheck = True
Caption = 'Align &Left'
Enabled = False
Hint = 'Align Left Aligns text at
the left indent'
ImageIndex = 35
end
V11V4

Caption = 'Standard Style'
GroupIndex = 1
OnExecute =
StdStyleActnExecute
end
object XPStyleActn: TAction
Category = 'Style'
AutoCheck = True
Caption = 'XP Style'
GroupIndex = 1
OnExecute =
XPStyleActnExecute
end
object ShadowActn: TAction
Category = 'Style'
AutoCheck = True
Caption = 'Menu Shadows'
OnExecute =
ShadowActnExecute
end
object FilePageSetup1:
TFilePageSetup
Category = 'File'
Caption = 'FilePageSetup1'
Dialog.MinMarginLeft = 0
Dialog.MinMarginTop = 0
Dialog.MinMarginRight = 0
Dialog.MinMarginBottom = 0
Dialog.MarginLeft = 1000
Dialog.MarginTop = 1000
Dialog.MarginRight = 1000
Dialog.MarginBottom = 1000
Dialog.PageWidth = 8500
Dialog.PageHeight = 11000
end
end
object ImageList1: TImageList
Left = 50
Top = 137
Bitmap = {
000000000000}
end
End
\\\\\\\\\

Caption = 'E&xit'
Hint = 'Exit Quits the
application'
ImageIndex = 43
end
object SearchFind1: TSearchFind
Category = 'Search'
Caption = '&Find'
Hint = 'Find Finds the specified
text'
ImageIndex = 34
ShortCut = 16454
end
object SearchFindNext1:
TSearchFindNext
Category = 'Search' Caption = 'Find &Next'
Enabled = False
Hint = 'Find Next Repeats the
last find'
ImageIndex = 33
ShortCut = 114
end
object SearchReplace1:
TSearchReplace
Category = 'Search'
Caption = '&Replace'
Hint = 'Replace Replaces specific
text with different text'
ImageIndex = 32
end
object SearchFindFirst1:
<b>TSearchFindFirst</b>
Category = 'Search'
Caption = 'F&ind First'
Hint = 'Find First Finds the first
occurance of specified text'
end
object CustomizeActionBars1:
TCustomizeActionBars
Category = 'Tools'
Caption = '&Customize'
CustomizeDlg.StayOnTop =
False
end
object StdStyleActn: TAction
Category = 'Style'
AutoCheck = True

Font.Height = -11
Font.Name = 'Tahoma'
Font.Style = []
Spacing = 4
end
object ToolActionBar1:
TActionToolBar
Left = 0
Top = 24
Width = 557
Height = 28
ActionManager =
ActionManager1
Caption = 'ToolActionBar1'
ColorMap.HighlightColor =
clBtnHighlight
ColorMap.UnusedColor =
15988985
ColorMap.SelectedColor =
clHighlight
EdgeBorders = [ebLeft, ebTop,
ebRight, ebBottom]
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
Spacing = 0
end
object ToolActionBar2:
TActionToolBar
Left = 0
Top = 52
Width = 557
Height = 28
ActionManager =
ActionManager1
Caption = 'ToolActionBar2'
ColorMap.HighlightColor =
clBtnHighlight
ColorMap.UnusedColor =
15988985
ColorMap.SelectedColor =
clHighlight
EdgeBorders = [ebLeft, ebTop,
ebRight, ebBottom]

الهلف التنفيذي لبرناهج
(ActionBands)
عداد
علاء الدين اللباد
الله ۱۷۹۵۸۵۶۹۰
ActionBands
object Form1: TForm1
Left = 202
Top = 114
Width = 565
Height = 368  Caption = 'ActionBars Demo'
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = False
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
object ActionMainMenuBar1:
<b>TActionMainMenuBar</b>
Left = 0
Top = 0
Width = 557
Height = 24
UseSystemFont = False
ActionManager =
ActionManager1 Caption = 'ActionMainMenuBar1'
ColorMap.HighlightColor =
clBtnHighlight
ColorMap.UnusedColor =
15988985
ColorMap.SelectedColor =
clHighlight
EdgeBorders = [ebBottom]
EdgeOuter = esNone
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText

XXV: 141 FFF
Width = 557
Height = 19
AutoHint = True
Panels = <>
end
object ActionManager1:
TActionManager
ActionBars.SessionCount = 6
ActionBars = <
item
Items = <
item
Items = <
item
Action = FileOpen1
ImageIndex = 30
ShortCut = 16463
end
item
Action = FileSaveAs1
ImageIndex = 8
end
item
Action = FileRun1
UsageCount = -1
end
item
Caption = '-'
end
item
Action = FileExit1
ImageIndex = 43
UsageCount = -1
end>
Caption = '&File'
end
item
Items = <
item
Action = EditCut1
ImageIndex = 0
ShortCut = 16472
end
item
Action = EditCopy1
ImageIndex = 1
ShortCut = 16451
end

Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
Spacing = 0
end
object ToolActionBar3:
TActionToolBar
Left = 0
Top = 80
Width = 557
Height = 28
ActionManager =
ActionManager1
Caption = 'ToolActionBar3'
ColorMap.HighlightColor =
clBtnHighlight
ColorMap.UnusedColor =
15988985
ColorMap.SelectedColor =
clHighlight
EdgeBorders = [ebLeft, ebTop,
ebRight, ebBottom]
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
Spacing = 0
end
object RichEdit1: TRichEdit
Left = 0
Top = 108
Width = 557
Height = 207
Align = alClient
BevelKind = bkTile
BorderStyle = bsNone
TabOrder = 4
end
object StatusBar1: TStatusBar
Left = 0
Top = 315
· · · · · · · · · · · · · · · · · · ·

,
end •
item
Action = RichEditItalic1
ImageIndex = 29
ShortCut = 16457
end
item
Action =
RichEditUnderline1
ImageIndex = 28
<b>ShortCut = 16469</b>
end
item
Action =
RichEditStrikeOut1
end
item
Action = RichEditBullets1
Caption = 'Bull&ets'
ImageIndex = 38
end
item
Action =
RichEditAlignLeft1
ImageIndex = 35
end
item
Action =
RichEditAlignRight1
ImageIndex = 36
end
item
Action =
RichEditAlignCenter1
ImageIndex = 37 end>
Caption = 'F&ormat'
end
item
Items = <
item
Action =
CustomizeActionBars1
UsageCount = -1
end>
Caption = '&Tools'
end
item

item
Action = EditPaste1
ImageIndex = 2
ShortCut = 16470
end
item
Action = EditSelectAll1
ShortCut = 16449
end
item
Action = EditUndo1
ImageIndex = 3
ShortCut = 16474
end
item
Action = EditDelete1
ImageIndex = 5
ShortCut = 46
end>
Caption = '&Edit'
end
item
Items = <
item
Action = SearchFind1
ImageIndex = 34
ShortCut = 16454
end
item
Action = SearchFindNext1
ImageIndex = 33
ShortCut = 114
end
item
Action = SearchReplace1
ImageIndex = 32
end
item
Action = SearchFindFirst1
end>
Caption = '&Search'
end
item
Items = <
item
Action = RichEditBold1
ImageIndex = 31
ShortCut = 16450
ShortCut - 10430

Caption = 'Bull&ets'
ImageIndex = 38
end
item
Action = RichEditAlignLeft1
ImageIndex = 35
end
item
Action =
RichEditAlignRight1
ImageIndex = 36
end
item
Action =
RichEditAlignCenter1
ImageIndex = 37
end>
ActionBar = ToolActionBar1
end
item
Items = <
item
Action = SearchFind1
ImageIndex = 34
ShortCut = 16454
end
item
Action = SearchFindNext1
ImageIndex = 33
ShortCut = 114
end item
Action = SearchReplace1
ImageIndex = 32
end
item
Caption = '-'
LastSession = 6
end
item
Action = SearchFindFirst1
end>
ActionBar = ToolActionBar2
AutoSize = False
end
item
Items = <
item
· · · · · · · · · · · · · · · · · · ·

T.
Items = <
item
Action = StdStyleActn
Caption = '&Standard
Style'
LastSession = 6
end
item
Action = XPStyleActn
Caption = '&XP Style'
LastSession = 6
end
item
Caption = '-'
end
item
Action = ShadowActn
Caption = '&Menu
Shadows'
LastSession = 6
end>
Caption = 'Sty≤'
end>
ActionBar =
ActionMainMenuBar1
end
item
Items = <
item
Action = RichEditBold1
ImageIndex = 31
ShortCut = 16450
end
item
Action = RichEditItalic1
ImageIndex = 29
ShortCut = 16457
end
item
Action = RichEditUnderline1
ImageIndex = 28
ShortCut = 16469
end
item
Action = RichEditStrikeOut1
end
item
Action = RichEditBullets1

Caption = '&Copy'
Enabled = False
Hint = 'Copy Copies the selection
and puts it on the Clipboard'
ImageIndex = 1
ShortCut = 16451
end
object EditPaste1: TEditPaste
Category = 'Edit'
Caption = '&Paste'
Enabled = False
Hint = 'Paste Inserts Clipboard
contents'
ImageIndex = 2
ShortCut = 16470
end
object EditSelectAll1:
TEditSelectAll
Category = 'Edit'
Caption = 'Select &All'
Enabled = False
Hint = 'Select All Selects the
entire document'
ShortCut = 16449
end
object EditUndo1: TEditUndo
Category = 'Edit'
Caption = '&Undo'
Enabled = False
Hint = 'Undo Reverts the last
action'
ImageIndex = 3
ShortCut = 16474
end
object EditDelete1: TEditDelete
Category = 'Edit'
Caption = '&Delete'
Enabled = False
Hint = 'Delete Erases the
selection'
ImageIndex = 5
ShortCut = 46
end
object RichEditBold1:
TRichEditBold
Category = 'Format'
AutoCheck = True
Caption = '&Bold'
Cupiton whole

A -4: E 1:4C41
Action = EditCut1
ImageIndex = 0
ShortCut = 16472
end
item
Action = EditCopy1
ImageIndex = 1
ShortCut = 16451
end
item
Action = EditPaste1
ImageIndex = 2
<b>ShortCut = 16470</b>
end
item
Action = EditSelectAll1
<b>ShortCut = 16449</b>
end
item
Action = EditUndo1
ImageIndex = 3
<b>ShortCut = 16474</b>
end
item
Action = EditDelete1
ImageIndex = 5
ShortCut = 46
end>
ActionBar = ToolActionBar3
AutoSize = False
end>
Images = ImageList1
OnStateChange =
ActionManager1StateChange
Left = 125
Top = 138
StyleName = 'Standard'
object EditCut1: TEditCut
Category = 'Edit'
Caption = 'Cu&t'
Enabled = False
Hint = 'Cut Cuts the selection
and puts it on the Clipboard'
ImageIndex = 0
ShortCut = 16472
end
object EditCopy1: TEditCopy
Category = 'Edit'

Enabled = False
Hint = 'Align Left Aligns text at
the left indent'
ImageIndex = 35
end
object RichEditAlignRight1:
TRichEditAlignRight
Category = 'Format'
AutoCheck = True
Caption = 'Align &Right'
Enabled = False
Hint = 'Align Right Aligns text at
the right indent'
ImageIndex = 36
end
object RichEditAlignCenter1:
TRichEditAlignCenter
Category = 'Format'
AutoCheck = True
Caption = '&Center'
Enabled = False
Hint = 'Center Centers text
between margins'
ImageIndex = 37
end
object FileOpen1: TFileOpen
Category = 'File'
Caption = '&Open'
Hint = 'Open Opens an existing
file'
ImageIndex = 30
ShortCut = 16463
OnAccept = FileOpen1Accept
end
object FileSaveAs1: TFileSaveAs
Category = 'File'
Caption = 'Save &As'
Hint = 'Save As Saves the active
file with a new name'
ImageIndex = 8
OnAccept = FileSaveAs1Accept
end
object FileRun1: TFileRun
Category = 'File'
Browse = False
BrowseDlg.Title = 'Run'
Caption = '&Run'
Hint = 'Run Runs an applicatoin'

Enabled = False
Hint = 'Bold'
ImageIndex = 31
ShortCut = 16450
end
object RichEditItalic1:
TRichEditItalic
Category = 'Format'
AutoCheck = True
Caption = '&Italic'
Enabled = False
Hint = 'Italic'
ImageIndex = 29
ShortCut = 16457
end
object RichEditUnderline1:
TRichEditUnderline
Category = 'Format'
AutoCheck = True
Caption = '&Underline'
Enabled = False
Hint = 'Underline'
ImageIndex = 28
ShortCut = 16469
end
object RichEditStrikeOut1:
TRichEditStrikeOut
Category = 'Format'
AutoCheck = True
Caption = '&Strikeout'
Enabled = False
Hint = 'Strikeout'
end
object RichEditBullets1:
TRichEditBullets
Category = 'Format'
AutoCheck = True
Caption = '&Bullets'
Enabled = False
Hint = 'Bullets Inserts a bullet on
the current line'
ImageIndex = 38
end
object RichEditAlignLeft1:
TRichEditAlignLeft
Category = 'Format'
AutoCheck = True
Caption = 'Align &Left'
Caption Angli Whete

False
end
object StdStyleActn: TAction
Category = 'Style'
AutoCheck = True
Caption = 'Standard Style'
GroupIndex = 1
OnExecute =
StdStyleActnExecute
end
object XPStyleActn: TAction
Category = 'Style'
AutoCheck = True
Caption = 'XP Style'
GroupIndex = 1
OnExecute =
XPStyleActnExecute
end
object ShadowActn: TAction
Category = 'Style'
AutoCheck = True
Caption = 'Menu Shadows'
OnExecute =
ShadowActnExecute
end
object FilePageSetup1:
TFilePageSetup
Category = 'File'
Caption = 'FilePageSetup1'
Dialog.MinMarginLeft = 0
Dialog.MinMarginTop = 0
Dialog.MinMarginRight = 0
Dialog.MinMarginBottom = 0
Dialog.MarginLeft = 1000
Dialog.MarginTop = 1000
Dialog.MarginRight = 1000
Dialog.MarginBottom = 1000
Dialog.PageWidth = 8500
Dialog.PageHeight = 11000
end
end
object ImageList1: TImageList
Left = 50
Top = 137
Bitmap = {
00000000000000}
end
End

Operation = 'open'
ShowCmd = scShowNormal
end
object FileExit1: TFileExit
Category = 'File'
Caption = 'E&xit'
Hint = 'Exit Quits the
application'
ImageIndex = 43
end
object SearchFind1: TSearchFind
Category = 'Search'
Caption = '&Find'
Hint = 'Find Finds the specified
text'
ImageIndex = 34
ShortCut = 16454
end
object SearchFindNext1:
TSearchFindNext
Category = 'Search'
Caption = 'Find &Next'
Enabled = False
Hint = 'Find Next Repeats the
last find'
ImageIndex = 33
ShortCut = 114
end
object SearchReplace1:
TSearchReplace
Category = 'Search'
Caption = '&Replace'
Hint = 'Replace Replaces specific
text with different text'
ImageIndex = 32
end
object SearchFindFirst1:
TSearchFindFirst
Category = 'Search'
Caption = 'F&ind First'
Hint = 'Find First Finds the first
occurance of specified text'
end
object CustomizeActionBars1:
TCustomizeActionBars
Category = 'Tools'
Caption = '&Customize'
CustomizeDlg.StayOnTop =

Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
StdCtrls, ExtCtrls, FrmData,
FrmMD;
type
TForm1 = class(TForm)
MDFrame: TMasterDetailFrame;
SimpleFrame: TDataFrame;
Splitter1: TSplitter;
<>procedure
FormCreate(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
enu,
WOM
Var
Form1: TForm1;
• •
implementation
(07)
{\$R *.dfm}
<>procedure
TForm1.FormCreate(Sender:
TObject);
begin
with SimpleFrame, Table1 do begin
TableName := 'Biolife';
with FancyFrame1 do begin
DBMemo1.DataSource :=
DataSource1;
DBMemo1.DataField := 'Notes';
DBImage1.DataSource :=
DataSource1;
DBImage1.DataField :=
'Graphic';
end;
Open;
end;
with MDFrame do begin
with MasterFrame, Table1 do
begin
FancyFrame1.Free;
TableName := 'Customer';
rabicivame Customer;

///////////////////////////////////////		

Procedure2
الهلف التنفيذي لبرناهج
(Db)
عاعد إ
علاء الدين اللباد
واتف ۷۳۹۵۷۵۶۹۰
علاء الدين اللباد ٢٩٥٢٥٥١، ٩٠
DB الملف التنفيذي للبرنامج
DB الملف التنفيذي للبرنامج يوجد فيه اجراءان فقط
Procedures 2
program FramesDemo;
uses
Forms,
FrmData in 'FrmData.pas'
{DataFrame: TFrame},
FrmFancy in 'FrmFancy.pas'
{FancyFrame: TFrame},
FrmMD in 'FrmMD.pas'
{MasterDetailFrame: TFrame},
FrmMain in 'FrmMain.pas'
{Form1};
{\$R *.RES}
begin
Application.Initialize;
Application.CreateForm(TForm1,
Form1);
Application.Run;
end.
unit FrmMain;
intonface
interface
waaa
uses

unit FrmData;
unit FimData,
interface
Interface
uses Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
ExtCtrls, DBCtrls, Db, DBTables,
Grids, DBGrids, FrmFancy;
type
TDataFrame = class(TFrame)
DBGrid1: TDBGrid;
DataSource1: TDataSource;
Table1: TTable;
DBNavigator1: TDBNavigator;
FancyFrame1: TFancyFrame;
Splitter1: TSplitter;
private deslarations)
{ Private declarations }
public
{ Public declarations }
end;
implementation
{\$R *.dfm}
end.
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
unit FrmFancy;
unit Frinfancy;
•
interface
W. J. W. C. W.
uses Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms,
Dialogs,
ExtCtrls, DBCtrls, StdCtrls;
type
TFancyFrame = class(TFrame)
DBMemo1: TDBMemo:

Open;			
end;			
,			
with DetailFrame, Table1 do			
begin			
FancyFrame1.Free;			
MasterSource :=			
MasterFrame.DataSource1;			
*			
MasterFields := 'CustNo';			
IndexName := 'CustNo';			
TableName := 'Orders';			
Open;			
•			
end;			
end;			
end;			
and			
end.			
\\\\\\\\			
unit FrmMD;			
unit Fimivid,			
interface			
ugas Windows Massass CII41			
TINGEN VALIFICATION OF INTERCOLOGICAL STATE OF THE			
uses Windows, Messages, SysUtils,			
Classes, Graphics, Controls, Forms,			
Classes, Graphics, Controls, Forms,			
Classes, Graphics, Controls, Forms, Dialogs,			
Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, FrmData;			
Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, FrmData; type			
Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, FrmData;  type TMasterDetailFrame =			
Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, FrmData;  type TMasterDetailFrame = class(TFrame)			
Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, FrmData;  type TMasterDetailFrame = class(TFrame)			
Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, FrmData;  type TMasterDetailFrame = class(TFrame) MasterFrame: TDataFrame;			
Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, FrmData;  type TMasterDetailFrame = class(TFrame) MasterFrame: TDataFrame; DetailFrame: TDataFrame;			
Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, FrmData;  type TMasterDetailFrame = class(TFrame) MasterFrame: TDataFrame; DetailFrame: TDataFrame; Splitter2: TSplitter;			
Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, FrmData;  type TMasterDetailFrame = class(TFrame) MasterFrame: TDataFrame; DetailFrame: TDataFrame;			
Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, FrmData;  type TMasterDetailFrame = class(TFrame) MasterFrame: TDataFrame; DetailFrame: TDataFrame; Splitter2: TSplitter; private			
Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, FrmData;  type TMasterDetailFrame = class(TFrame) MasterFrame: TDataFrame; DetailFrame: TDataFrame; Splitter2: TSplitter; private { Private declarations }			
Classes, Graphics, Controls, Forms, Dialogs,  ExtCtrls, FrmData;  type  TMasterDetailFrame = class(TFrame)  MasterFrame: TDataFrame;  DetailFrame: TDataFrame;  Splitter2: TSplitter;  private { Private declarations }  public			
Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, FrmData;  type TMasterDetailFrame = class(TFrame) MasterFrame: TDataFrame; DetailFrame: TDataFrame; Splitter2: TSplitter; private { Private declarations } public { Public declarations }			
Classes, Graphics, Controls, Forms, Dialogs,  ExtCtrls, FrmData;  type  TMasterDetailFrame = class(TFrame)  MasterFrame: TDataFrame;  DetailFrame: TDataFrame;  Splitter2: TSplitter;  private { Private declarations }  public			
Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, FrmData;  type TMasterDetailFrame = class(TFrame) MasterFrame: TDataFrame; DetailFrame: TDataFrame; Splitter2: TSplitter; private { Private declarations } public { Public declarations }			
Classes, Graphics, Controls, Forms, Dialogs,  ExtCtrls, FrmData;  type  TMasterDetailFrame = class(TFrame)  MasterFrame: TDataFrame;  DetailFrame: TDataFrame;  Splitter2: TSplitter; private { Private declarations } public { Public declarations } end;			
Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, FrmData;  type TMasterDetailFrame = class(TFrame) MasterFrame: TDataFrame; DetailFrame: TDataFrame; Splitter2: TSplitter; private { Private declarations } public { Public declarations }			
Classes, Graphics, Controls, Forms, Dialogs,  ExtCtrls, FrmData;  type  TMasterDetailFrame = class(TFrame)  MasterFrame: TDataFrame;  DetailFrame: TDataFrame;  Splitter2: TSplitter;  private { Private declarations }  public { Public declarations }  end;  implementation			
Classes, Graphics, Controls, Forms, Dialogs,  ExtCtrls, FrmData;  type  TMasterDetailFrame = class(TFrame)  MasterFrame: TDataFrame;  DetailFrame: TDataFrame;  Splitter2: TSplitter; private { Private declarations } public { Public declarations } end;			
Classes, Graphics, Controls, Forms, Dialogs,  ExtCtrls, FrmData;  type  TMasterDetailFrame = class(TFrame)  MasterFrame: TDataFrame;  DetailFrame: TDataFrame;  Splitter2: TSplitter;  private { Private declarations }  public { Public declarations }  end;  implementation			
Classes, Graphics, Controls, Forms, Dialogs,  ExtCtrls, FrmData;  type  TMasterDetailFrame = class(TFrame)  MasterFrame: TDataFrame;  DetailFrame: TDataFrame;  Splitter2: TSplitter;  private { Private declarations }  public { Public declarations }  end;  implementation			
Classes, Graphics, Controls, Forms, Dialogs,  ExtCtrls, FrmData;  type  TMasterDetailFrame = class(TFrame)  MasterFrame: TDataFrame;  DetailFrame: TDataFrame;  Splitter2: TSplitter;  private { Private declarations }  public { Public declarations }  end;  implementation			
Classes, Graphics, Controls, Forms, Dialogs,  ExtCtrls, FrmData;  type  TMasterDetailFrame = class(TFrame)  MasterFrame: TDataFrame;  DetailFrame: TDataFrame;  Splitter2: TSplitter;  private { Private declarations }  public { Public declarations }  end;  implementation			
Classes, Graphics, Controls, Forms, Dialogs,  ExtCtrls, FrmData;  type  TMasterDetailFrame = class(TFrame)  MasterFrame: TDataFrame;  DetailFrame: TDataFrame;  Splitter2: TSplitter;  private { Private declarations }  public { Public declarations }  end;  implementation			

end
inline MDFrame:
TMasterDetailFrame
Left = 0
Top = 297
Width = 775
Height = 263
Align = alBottom
TabOrder = 0
inherited Splitter2: TSplitter
Height = 263
end
inherited MasterFrame:
<b>TDataFrame</b>
Height = 263
inherited Splitter1: TSplitter
Top = 159
end
inherited DBGrid1: TDBGrid
Height = 134
end
inherited DBNavigator1:
TDBNavigator
Hints.Strings = ()
end
inherited FancyFrame1:
TFancyFrame
Top = 162
end
end
inherited DetailFrame:
<b>TDataFrame</b>
Width = 434
Height = 263
inherited Splitter1: TSplitter
Top = 159
Width = 434
end
inherited DBGrid1: TDBGrid
Width = 434
Height = 134
inherited DBNavigator1:
TDBNavigator
Width = 434
Hints.Strings = ()
end
inherited FancyFrame1:

DBImage1: TDBImage;		
Splitter1: TSplitter;		
private		
{ Private declarations }		
public		
{ Public declarations }		
end;		
implementation		
{\$R *.dfm}		
end.		
111111111111111111111111111111111111111		

Objects 15		
الهلف النصي لبرناهج		
(DB)		
عامدا		
علاء الدين اللباد		
واتف ۷۹۲۵۷۵۲۹۰		
object Form1: TForm1		
Left = 209		
Top = 158		
Width = 783		
Height = 594		
Caption = 'Form1'		
Color = clBtnFace		
ParentFont = True		
OldCreateOrder = False		
Position = poScreenCenter		
OnCreate = FormCreate		
PixelsPerInch = 96		
TextHeight = 13		
<> object Splitter1: TSplitter		
Left = 0		
Top = 294		
Width = 775		
Height = 3		
Cursor = crVSplit		
Align = alBottom		

object MasterDetailFrame:
TMasterDetailFrame
Left = 0
Top = 0
Width = 553
Height = 367
TabOrder = 0
<> object Splitter2: TSplitter
Left = 338
Top = 0
Height = 367
end
inline MasterFrame: TDataFrame
Left = 0
Top = 0
Width = 338
Height = 367
Align = alLeft
TabOrder = 0
inherited Splitter1: TSplitter
Top = 263
end
inherited DBGrid1: TDBGrid
Height = 238
end
inherited DBNavigator1:
TDBNavigator
Hints.Strings = ()
end
inherited FancyFrame1:
TFancyFrame
Top = 266
end
end
inline DetailFrame: TDataFrame
Left = 341
Top = 0 Width = 212
Height = 367
Align = alClient
TabOrder = 1
inherited Splitter1: TSplitter
Top = 263
Width = 212
end
inherited DBGrid1: TDBGrid
Width = 212
Height = 238

TFancyFrame		
Top = 162		
Width = 434		
inherited DBImage1:		
TDBImage		
Width = 270		
end		
inline SimpleFrame: TDataFrame		
Left = 0		
Top = 0		
Width = 775		
Height = 294		
Align = alClient		
TabOrder = 1		
inherited Splitter1: TSplitter		
Top = 190		
Width = 775		
end		
inherited DBGrid1: TDBGrid		
Width = 775		
Height = 165		
end		
inherited DBNavigator1:		
TDBNavigator		
Width = 775		
Hints.Strings = ()		
end		
inherited FancyFrame1:		
TFancyFrame		
Top = 193		
Width = 775		
inherited Splitter1: TSplitter		
Left = 489		
end		
inherited DBMemo1: TDBMemo		
Width = 489		
end		
inherited DBImage1: TDBImage		
Left = 492		
Width = 283		
end		

Left = 0
Top = 0
Width = 338
Height = 25
DataSource = DataSource1
Align = alTop
TabOrder = 1
end
inline FancyFrame1: TFancyFrame
Left = 0
Top = 260
Width = 338
Height = 101
Align = alBottom
TabOrder = 2
inherited DBImage1: TDBImage
Width = 174
end
end
<> object DataSource1:
TDataSource
DataSet = Table1
Left = 40
Top = 200
end
<> object Table1: TTable
DatabaseName = 'DBDEMOS'
Left = 8
Top = 200
end
End
object FancyFrame: TFancyFrame
Left = 0
Top = 0
Width = 320
Height = 101
TabOrder = 0
<> object Splitter1: TSplitter
Left = 161
Top = 0
Height = 101
end
<> object DBMemo1:
TDBMemo
Left = 0
Top = 0
Width = 161
Height = 101
iivigiit ivi

end
inherited DBNavigator1:
TDBNavigator
Width = 212
Hints.Strings = ()
end
inherited FancyFrame1:
TFancyFrame
Top = 266
Width = 212
inherited DBImage1: TDBImage
Width = 48
end
end
end
End
Litt
object DataFrame: TDataFrame
Left = 0
Top = 0 Width = 338
Height = 361
TabOrder = 0
<pre>&lt;&gt; object Splitter1: TSplitter</pre>
Left = 0
Top = 257
Width = 338
Height = 3
Cursor = crVSplit
Align = alBottom
end
<> object DBGrid1: TDBGrid
Left = 0
Top = 25
Width = 338
Height = 232
Align = alClient
DataSource = DataSource1
TabOrder = 0
TitleFont.Charset =
DEFAULT CHARSET
TitleFont.Color = clWindowText
TitleFont.Height = -11
TitleFont.Name = 'MS Sans Serif'
TitleFont.Style = []
end
<pre>&lt;&gt; object DBNavigator1:</pre>
<b>TDBNavigator</b>

D4TTD44
BtnTrans: TButton;
<> procedure
BtnCloseClick(Sender: TObject);
<> procedure
BtnViewsClick(Sender: TObject);
<> procedure
BtnTriggClick(Sender: TObject);
<> procedure
BtnQrySPClick(Sender: TObject);
<> procedure
BtnExecSPClick(Sender: TObject);
<> procedure
BtnTransClick(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
FrmLauncher: TFrmLauncher;
,
implementation
•
{\$R *.dfm}
uses
FrmExeSP, { The Executable
Stored Procedure Demo }
FrmQrySP, { The Query Stored
Procedure demo }
FrmTrans, { The Transaction
Editing demo }
FrmTrigg, { The Trigger Demo }
FrmViews, { The View Demo }
Variants:
v ariants;
procedure
TFrmLauncher.BtnCloseClick(Send
er: TObject);
begin
Close;
end;
procedure
TFrmLauncher.BtnViewsClick(Send
er: TObject);
begin

Align = alLeft
TabOrder = 0
end
<> object DBImage1:
TDBImage
Left = 164
Top = 0
Width = 156
Height = 101
Align = alClient
TabOrder = 1
end
End

الملف التنفيذي للبرنامج ويحتوي على ٣٢ إجراءا (Procedure32) ويعتبر من البرامج الشاملة المرتبطة بقواعد البيانات وهو برنامج مفتوح الشيفرة وموجود ضمن دلفي ٧ باسم CsDemos

Procedure32					
المك التنفيذي لبرنامج (CsDemos) إعداد					
			علاء الدين اللباد		
			واتف ۷۳۵۷۵۶۹۰		
unit Frmmain;					
unit Frimmani,					
interface					
uses					
SysUtils, Windows, Messages,					
Classes, Graphics, Controls,					
Forms, Dialogs, DB, StdCtrls;					
type					
TFrmLauncher = class(TForm)					
BtnTrigg: TButton;					
BtnViews: TButton;					
BtnQrySP: TButton;					
BtnExecSP: TButton;					
BtnClose: TButton;					

alass(TDataMadula)	FrmViewDemo.ShowModal;
class(TDataModule)	,
SalesTable: TTable;	end;
SalesSource: TDataSource;	
CustomerTable: TTable;	procedure
CustomerSource: TDataSource;	TFrmLauncher.BtnTriggClick(Send
ShipOrderProc: TStoredProc;	er: TObject);
EmployeeDatabase: TDatabase;	begin
EmployeeSource: TDataSource;	FrmTriggerDemo.ShowModal;
EmployeeTable: TTable;	end;
EmployeeTableEMP_NO:	
TSmallintField;	procedure
EmployeeTableFIRST_NAME:	TFrmLauncher.BtnQrySPClick(Sen
TStringField;	der: TObject);
EmployeeTableLAST_NAME:	begin
TStringField;	FrmQueryProc.ShowModal;
EmployeeTablePHONE_EXT:	end;
TStringField;	
EmployeeTableHIRE_DATE:	procedure
TDateTimeField;	TFrmLauncher.BtnExecSPClick(Se
EmployeeTableDEPT_NO:	nder: TObject);
TStringField;	begin
EmployeeTableJOB_CODE:	FrmExecProc.ShowModal;
TStringField;	end;
EmployeeTableJOB_GRADE:	
TSmallintField;	procedure
EmployeeTableJOB_COUNTRY:	TFrmLauncher.BtnTransClick(Send
TStringField;	er: TObject);
EmployeeTableSALARY:	begin
TFloatField;	FrmTransDemo.ShowModal;
EmployeeTableFULL_NAME:	end;
TStringField;	
DeleteEmployeeProc:	end.
TStoredProc;	
SalaryHistoryTable: TTable;	
SalaryHistorySource:	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
TDataSource;	
	unit DmCSDemo;
SalaryHistoryTableCHANGE_DAT	
E: TDateTimeField;	interface
SalaryHistoryTableUPDATER_ID:	uses
TStringField;	Windows, Messages, SysUtils,
	Classes, Graphics, Controls, Forms,
SalaryHistoryTableOLD_SALARY:	Dialogs,
TFloatField;	DBTables, DB;
SalaryHistoryTablePERCENT_CH	type
ANGE: TFloatField;	TDmEmployee =
1	I - J

TIntegerField;	SalaryHistoryTableNEW SALARY:
SalesTableDISCOUNT:	TFloatField;
TFloatField;	CustomerTableCUST_NO:
SalesTableITEM TYPE:	TIntegerField;
TStringField;	CustomerTableCUSTOMER:
SalesTableAGED: TFloatField;	TStringField;
EmployeeLookup: TTable;	
SmallintField1: TSmallintField;	CustomerTableCONTACT_FIRST:
StringField1: TStringField;	TStringField;
StringField2: TStringField;	
StringField3: TStringField;	CustomerTableCONTACT_LAST:
DateTimeField1:	TStringField;
TDateTimeField;	CustomerTablePHONE_NO:
StringField4: TStringField;	TStringField;
StringField5: TStringField;	
SmallintField2: TSmallintField;	CustomerTableADDRESS_LINE1:
StringField6: TStringField;	TStringField;
FloatField1: TFloatField;	
StringField7: TStringField;	<b>CustomerTableADDRESS_LINE2:</b>
SalaryHistoryTableEMP NO:	TStringField;
TSmallintField;	CustomerTableCITY:
SalaryHistoryTableEMPLOYEE:	TStringField;
TStringField;	
<> procedure	CustomerTableSTATE_PROVINCE
EmployeeTableBeforeDelete(DataSe	: TStringField;
t: TDataSet);	CustomerTableCOUNTRY:
<> procedure	TStringField;
EmployeeTableAfterPost(DataSet:	CustomerTablePOSTAL_CODE:
TDataSet);	TStringField;
<> procedure	CustomerTableON_HOLD:
DmEmployeeCreate(Sender:	TStringField;
TObject);	SalesTablePO_NUMBER:
private	TStringField;
{ Private declarations }	SalesTableCUST_NO:
public	TIntegerField;
{ Public declarations }	SalesTableSALES_REP:
end;	TSmallintField;
	SalesTableORDER_STATUS:
var	TStringField;
DmEmployee: TDmEmployee;	SalesTableORDER_DATE:
	TDateTimeField;
implementation	SalesTableSHIP_DATE:
	TDateTimeField;
{\$R *.dfm}	SalesTableDATE_NEEDED:
	TDateTimeField;
{ Note: Business rules go in the data	SalesTablePAID: TStringField;
model. Here is an example, used by	SalesTableQTY_ORDERED:
the transaction editing demo.	TIntegerField;
	SalesTableTOTAL_VALUE:

with SalaryHistoryTable do if Active then Refresh;	Deletes for the employee table are done
end;	with a stored procedure rather
enu,	than the normal BDE record delete
procedure	mechanism, so an audit trail could
-	be provided, etc }
TDmEmployee.DmEmployeeCreate( Sender: TObject);	be provided, etc }
begin	{ The database, EmployeeDatabase,
EmployeeDatabase.Open;	is the InterBase example
	EMPLOYEE.GDB database
end;	accessed thru the BDE alias
and a	IBLOCAL. This database contains
end.	examples
	of stored procedures, triggers,
111111111111111111111111111111111111111	check constraints, views, etc., many
	of
unit Frmexesp;	which are used within this demo
	project. }
interface	project. 3
	procedure
uses	TDmEmployee.EmployeeTableBefor
SysUtils, Windows, Messages,	eDelete(DataSet: TDataSet);
Classes, Graphics, Controls,	begin
StdCtrls, Forms, DBCtrls, DB,	{ Assign the current employee's id
DBGrids, Buttons, DBTables, Mask,	to the stored procedure's parameter
Grids,	}
ExtCtrls;	
4	DeleteEmployeeProc.Params.Param
TError Even Property along (TError)	Values['EMP NUM'] :=
TFrmExecProc = class(TForm)  DBGrid1: TDBGrid;	EmployeeTable['EMP_NO'];
,	DeleteEmployeeProc.ExecProc;
ScrollBox: TScrollBox;	{ Trigger the stored proc }
Label1: TLabel;	EmployeeTable.Refresh; {
EditCUST_NO: TDBEdit;	Refresh the data }
Label2: TLabel;	{ Block the EmployeeTable delete
EditCUSTOMER: TDBEdit;	since the stored procedure did the
Label3: TLabel;	work }
EditCONTACT_FIRST:	Abort;
TDBEdit;	end;
EditCONTACT_LAST: TDBEdit;	
EditPHONE NO: TDBEdit;	procedure
Label6: TLabel;	TDmEmployee.EmployeeTableAfter
EditADDRESS_LINE: TDBEdit;	Post(DataSet: TDataSet);
EditADDRESS LINE2: TDBEdit;	begin
EditCITY: TDBEdit;	{ A change in an employee salary
EditSTATE_PROVINCE:	triggers a change in the salary
TDBEdit;	history,
EditCOUNTRY: TDBEdit;	so if that table is open, it needs to
EditPOSTAL_CODE: TDBEdit;	be refreshed now }

TObject);	DBNavigator: TDBNavigator;
• • • • • • • • • • • • • • • • • • • •	Panel1: TPanel;
begin (Di 11 D / F / f	,
{ Disable DataEvents from the	Panel3: TPanel;
SalesTable for this form now }	Panel2: TPanel;
SalesSource.Enabled := False;	DBCheckBox1: TDBCheckBox;
end;	Label4: TLabel;
	BtnShipOrder: TSpeedButton;
procedure	BitBtn1: TBitBtn;
TFrmExecProc.SalesSourceDataCha	SalesSource: TDataSource;
nge(Sender: TObject; Field: TField);	<> procedure
begin	SalesSourceDataChange(Sender:
if	TObject; Field: TField);
DmEmployee.SalesTable['ORDER	<> procedure
STATUS'] <> NULL then	BtnShipOrderClick(Sender:
BtnShipOrder.Enabled :=	TObject);
p 0.1 west to a west to	<> procedure
AnsiCompareText(DmEmployee.Sal	FormShow(Sender: TObject);
esTable['ORDER STATUS'],	<> procedure
'SHIPPED') <> 0;	-
end;	FormHide(Sender: TObject);
enu;	private
1	{ private declarations }
procedure	public
TFrmExecProc.BtnShipOrderClick(	{ public declarations }
Sender: TObject);	end;
begin	
with DmEmployee do	var
begin	FrmExecProc: TFrmExecProc;
ShipOrderProc.Params[0].AsString	implementation
:= SalesTable['PO_NUMBER'];	
ShipOrderProc.ExecProc;	uses DmCSDemo, Variants;
SalesTable.Refresh;	
end;	{\$R *.dfm}
end;	(#11 14111)
	procedure
end.	TFrmExecProc.FormShow(Sender:
	TObject);
	9 //
	begin  DmEmployee SalesTable Open
	DmEmployee.SalesTable.Open;
*4 E	DmEmployee.CustomerTable.Open;
unit Frmmain;	{ Enable DataEvents from the
	SalesTable for this form now }
interface	SalesSource.Enabled := True;
	end;
uses	
SysUtils, Windows, Messages,	procedure
Classes, Graphics, Controls,	TFrmExecProc.FormHide(Sender:
, , , , , , , , , , , , , , , , , , , ,	

ov. TObicat):	Forms Dialogs DR StdCtrls.
er: TObject);	Forms, Dialogs, DB, StdCtrls;
Clare	4vm o
Close;	type
end;	TFrmLauncher = class(TForm)
_	BtnTrigg: TButton;
procedure	BtnViews: TButton;
TFrmLauncher.BtnViewsClick(Send	BtnQrySP: TButton;
er: TObject);	BtnExecSP: TButton;
begin	BtnClose: TButton;
FrmViewDemo.ShowModal;	BtnTrans: TButton;
end;	<> procedure
	BtnCloseClick(Sender: TObject);
procedure	<> procedure
TFrmLauncher.BtnTriggClick(Send	BtnViewsClick(Sender: TObject);
er: TObject);	<> procedure
begin	BtnTriggClick(Sender: TObject);
FrmTriggerDemo.ShowModal;	<> procedure
end;	BtnQrySPClick(Sender: TObject);
	<> procedure
procedure	BtnExecSPClick(Sender: TObject);
TFrmLauncher.BtnQrySPClick(Sen	<> procedure
der: TObject);	BtnTransClick(Sender: TObject);
begin	private
FrmQueryProc.ShowModal;	{ Private declarations }
end;	public
- Chu,	{ Public declarations }
procedure	end;
TFrmLauncher.BtnExecSPClick(Se	cird,
nder: TObject);	var
begin	FrmLauncher: TFrmLauncher;
FrmExecProc.ShowModal;	Timeauncher, Trimeauncher,
end;	implementation
Citu,	implementation
procedure	(CD + 15)
procedure TErmI aunahar PtnTransClick(Sand	{\$R *.dfm}
TFrmLauncher.BtnTransClick(Send	
er: TObject);	uses
begin  ErmTransDama ShawMadala	FrmExeSP, { The Executable
FrmTransDemo.ShowModal;	Stored Procedure Demo }
end;	FrmQrySP, { The Query Stored
	Procedure demo }
end.	FrmTrans, { The Transaction
	Editing demo }
	FrmTrigg, { The Trigger Demo }
	FrmViews, { The View Demo }
	Variants;
unit Frmqrysp;	
	procedure
interface	TFrmLauncher.BtnCloseClick(Send

TFrmQueryProc.WriteMsg(StrWrit e: String); uses	
e: orring):	
begin SysUtils, Windows, Messages,	
StatusBar1.SimpleText := Classes, Graphics, Controls,	
StrWrite; StdCtrls, Forms, DBCtrls, DB,	
end; DBGrids, DBTables, Grids,	
ExtCtrls,	
procedure Dialogs, Buttons, Mask, ComC	trls;
TFrmQueryProc.FormShow(Sender	
: TObject); type	
begin TFrmQueryProc = class(TForm	n)
DBGrid1: TDBGrid;	
DmEmployee.EmployeeTable.Open; DBGrid2: TDBGrid;	
{ Allow data flow from the DBNavigator: TDBNavigator:	
EmployeeTable to the local Panel1: TPanel;	,
EmployeeSource. This  Panel2: TPanel;	
will allow DataChange events to  Panel3: TPanel;	
execute the query procedure }  EmployeeProjectsQuery: TQu	lerv.
EmployeeSource.Enabled := True; EmployeeProjectsSource:	101 y ,
{ Explicit query preparation is not TDataSource:	
required, but gives the best possible  StatusBar1: TStatusBar;	
Diebent, 15teben,	
Employeesoureet I Buttusour	ce;
procedure	
Employee Data Change (Senaci:	
TObject; Field: TField);	
procedure <> procedure	
TFrmQueryProc.EmployeeDataCha FormShow(Sender: TObject);	
nge(Sender: TObject; Field: TField); <> procedure	
begin FormHide(Sender: TObject);	
{ Execute the ProjectsQuery, which private	
uses a query procedure } { private declarations }	
EmployeeProjectsQuery.Close; <> procedure writeMsg(	
strWrite: String);	
EmployeeProjectsQuery.Params[0]. public	
AsInteger := { public declarations }	
end;	
DmEmployee.EmployeeTableEmp_	
No.Value; var	
EmployeeProjectsQuery.Open; FrmQueryProc: TFrmQueryProc	roc:
	7
WriteMsg('Employee ' + implementation	
DmEmployee.EmployeeTableEmp_	
No.AsString + uses DmCSDemo;	
' is assigned to ' +	
IntToStu(EmployeeDucioatsQueny D	
ecordCount) + {\$R *.dfm}	
'nraject(s) ')·	
project(s). ); procedure	

{ private declarations }	
public	procedure
{ public declarations }	TFrmQueryProc.FormHide(Sender:
end;	TObject);
	begin
var	{ Turn off the DataChange event
FrmTransDemo:	for our form, since
TFrmTransDemo;	DmEmployee.EmployeeTable
·	is used elsewhere }
implementation	EmployeeSource.Enabled := False;
	end;
uses DmCSDemo;	
	end.
{\$R *.dfm}	
	111111111111111111111111111111111111111
procedure	
TFrmTransDemo.FormShow(Sende	unit Frmtrans;
r: TObject);	
begin	interface
DmEmployee.EmployeeDatabase.St	uses
artTransaction;	SysUtils, Windows, Messages,
	Classes, Graphics, Controls,
DmEmployee.EmployeeTable.Open;	StdCtrls, Forms, DBCtrls, DB,
end;	DBGrids, Buttons, DBTables, Grids,
	ExtCtrls,
procedure	Dialogs, BDE;
TFrmTransDemo.FormHide(Sender	
: TObject);	type
begin	TFrmTransDemo = class(TForm)
	DBGrid1: TDBGrid;
DmEmployee.EmployeeDatabase.Co	DBNavigator: TDBNavigator;
mmit;	Panel1: TPanel;
end;	Panel2: TPanel;
nuo coduno	BitBtn1: TBitBtn;
procedure TFrmTransDemo.BtnCommitEdits	BtnUndoEdits: TSpeedButton;
Click(Sender: TObject);	BtnCommitEdits: TSpeedButton;
begin	<pre>&gt; procedure Pto Committe dita Clinic (Soundary)</pre>
if	BtnCommitEditsClick(Sender:
DmEmployee.EmployeeDatabase.In	TObject);
Transaction and	<pre>&lt;&gt; procedure RtnUndoEditsClick(Sondor:</pre>
(MessageDlg('Are you sure you	BtnUndoEditsClick(Sender: TObject);
want to commit your changes?',	<> procedure
mtConfirmation, [mbYes,	FormShow(Sender: TObject);
mbNo , 0) = mrYes) then	<> procedure
begin	FormHide(Sender: TObject);
~~~	private
DmEmployee.EmployeeDatabase.Co	private
proj ttt2proj tt2	

HEAC	mmit;
SysUtils, Windows, Messages,	11111111,
Classes, Graphics, Controls,	DmEmployee.EmployeeDatabase.St
StdCtrls, Forms, DBCtrls, DB,	artTransaction;
DBGrids, Buttons, DBTables, Grids,	artiransaction,
ExtCtrls;	DmEmployee.EmployeeTable.Refres
Externs,	h;
trino.	end else
type TFrmTriggerDemo = class(TForm)	
DBGrid1: TDBGrid;	MessageDlg('Can''t Commit Changes: No Transaction Active',
,	mtError, [mbOk], 0);
DBGrid2: TDBGrid;	end;
DBNavigator: TDBNavigator;	enu;
Panel1: TPanel;	nuaaaduua
Panel2: TPanel;	procedure TFrmTransDemo.BtnUndoEditsClic
Panel3: TPanel;	
BitBtn1: TBitBtn;	k(Sender: TObject);
<> procedure	begin if
FormShow(Sender: TObject);	DmEmployee.EmployeeDatabase.In
private	Transaction and
{ private declarations }	(MessageDlg('Are you sure you
public	want to undo all changes made
{ public declarations }	during the '+
end;	'current transaction?',
	mtConfirmation, [mbYes, mbNo], 0)
var	= mrYes) then
FrmTriggerDemo:	begin
TFrmTriggerDemo;	begin
	DmEmployee.EmployeeDatabase.Ro
implementation	llback;
D. COD	110 110 110
uses DmCSDemo;	DmEmployee.EmployeeDatabase.St
(OD 4, 16 )	artTransaction;
{\$R *.dfm}	,
	DmEmployee.EmployeeTable.Refres
procedure	h;
TFrmTriggerDemo.FormShow(Send	end else
er: TObject);	MessageDlg('Can''t Undo Edits:
begin	No Transaction Active', mtError,
Des Employee Employee Table On	[mbOk], 0);
DmEmployee.EmployeeTable.Open;	end;
DmEmpleyee SalamyHistory-Table O	
DmEmployee.SalaryHistoryTable.O	end.
pen;	
end;	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
and a	unit Frmtrigg;
end.	<b>39</b> /
	interface
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	

procedure
TFrmViewDemo.ShowTable(
ATable: string );
begin
Screen.Cursor := crHourglass; {
show user something's happening }
VaryingTable.DisableControls; {
hide data changes from user }
VaryingTable.Active := FALSE;
{ close the table }
VaryingTable.TableName :=
ATable; { update the name }
VaryingTable.Open; {
open the table }
VaryingTable.EnableControls;
{ paint the changes }
Screen.Cursor := crDefault; {
reset the pointer }
end;
procedure
TFrmViewDemo.FormShow(Sender
: TObject);
begin
VaryingTable.Open;
end;
procedure
TFrmViewDemo.BtnShowEmployee
Click(Sender: TObject);
begin
ShowTable('EMPLOYEE');
end;
,
procedure
TFrmViewDemo.BtnShowPhoneList
Click(Sender: TObject);
begin
ShowTable('PHONE LIST');
end;
,
end.
\\\\\\\

unit Frmviews;
unit Frmviews;
interface
MSOS
Systicis Windows Mossoges
SysUtils, Windows, Messages, Classes, Graphics, Controls,
StdCtrls, Forms, DBCtrls, DB,
DBGrids, Buttons, DBTables, Grids,
ExtCtrls;
type
TFrmViewDemo = class(TForm)
DBGrid1: TDBGrid;
DBNavigator: TDBNavigator;
Panel1: TPanel;
VaryingTableSource:
TDataSource;
Panel2: TPanel;
VaryingTable: TTable;
BitBtn1: TBitBtn;
BtnShowEmployee:
TSpeedButton;
BtnShowPhoneList:
TSpeedButton;
<> procedure
BtnShowEmployeeClick(Sender:
TObject);
<pre>&lt;&gt; procedure BtnShowPhoneListClick(Sender:</pre>
TObject);
<> procedure
FormShow(Sender: TObject);
private
{ private declarations }
<> procedure
ShowTable(ATable: string);
public
{ public declarations }
end;
var
FrmViewDemo: TFrmViewDemo;
Tim view Demo. If im view Demo;
implementation
{\$R *.dfm}

DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
ShowHint = True
OnShow = FormShow
PixelsPerInch = 96
TextHeight = 13
<>object Panel1: TPanel
Left = 0
Top = 0
Width = 390
Height = 41
Align = alTop
TabOrder = 0
<>object BtnShowEmployee:
TSpeedButton
Left = 258
Top = 8
Width = 25
Height = 25
Hint = 'Show Employee Table'
GroupIndex = 1
Down = True
Glyph.Data = {
777777}
NumGlyphs = 2
OnClick =
BtnShowEmployeeClick
end
<>object
BtnShowPhoneList: TSpeedButton
Left = 282
Top = 8
Width = 25
Height = 25
Hint = 'Show Phone List View'
GroupIndex = 1
-
Glyph.Data = {
7777707EE077777
77777707FF0777777}
NumGlyphs = 2
OnClick =
BtnShowPhoneListClick

ObjectS
الملف النصي لبرنامج
(CsDemos)
عداد
علاء الدين اللباد
واتف ۷۱۳۵۷۵۶۹۰
Objects 139
<>object FrmViewDemo:
TFrmViewDemo
Left = 298
Top = 177
Width = 398
<b>Height = 332</b>
ActiveControl = Panel1
Caption = 'Table/View Switcher'
Color = clBtnFace
Font.Charset =
L

THE COLUMN IN
TitleFont.Color = clBlack
TitleFont.Height = -11
TitleFont.Name = 'MS Sans
Serif'
TitleFont.Style = []
end
end
<>object
VaryingTableSource: TDataSource
DataSet = VaryingTable
Left = 52
Top = 135
end
<>object VaryingTable:
TTable
DatabaseName =
'EmployeeDemoDB'
TableName = 'EMPLOYEE'
Left = 169
Top = 119
end
End
\\\\\
<>object FrmTriggerDemo:
TFrmTriggerDemo
Left = 684
Top = 520
Width = 391
Height = 342
<b>Hint = 'Explore the EmployeeTable</b>
to see the trigger sources'
ActiveControl = Panel1
Caption = 'Salary Change
Auditing'
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
ShowHint = True
OnShow = FormShow
PixelsPerInch = 96

end
<>object DBNavigator:
TDBNavigator
Left = 8
Top = 8
Width = 240
Height = 25
DataSource =
VaryingTableSource
Ctl3D = False
ParentCtl3D = False
TabOrder = 0
end
<>object BitBtn1: TBitBtn
Left = 320
Top = 8
Width = 60
Height = 25
Hint = 'Exit and close this form'
Caption = 'E&xit'
TabOrder = 1
Kind = bkClose
Style = bsNew
end
end
<>object Panel2: TPanel
Left = 0
Top = 41
Width = 390
Height = 257
Align = alClient
BevelInner = bvLowered
BorderWidth = 4
Caption = 'Panel2'
TabOrder = 1
<>object DBGrid1:
TDBGrid
Left = 6
Top = 6
Width = 378
Height = 245
Align = alClient
BorderStyle = bsNone
DataSource =
VaryingTableSource
TabOrder = 0
TitleFont.Charset =
DEFAULT_CHARSET

The second secon
Top = 6
Width = 371
Height = 125
Hint = 'Changing a salary
initiates a trigger'
Align = alClient
BorderStyle = bsNone
DataSource =
DmEmployee.EmployeeSource
TabOrder = 0
TitleFont.Charset =
DEFAULT_CHARSET
TitleFont.Color = clBlack
TitleFont.Height = -11
TitleFont.Name = 'MS Sans
Serif'
TitleFont.Style = []
Columns = <
item
Expanded = False
FieldName = 'FULL_NAME'
ReadOnly = True
Width = 124
Visible = True
end
item
Expanded = False
FieldName = 'SALARY'
Visible = True
end
item
Expanded = False
FieldName = 'JOB_GRADE'
ReadOnly = True
Visible = True
end>
end
end
<>object Panel3: TPanel
Left = 0
Top = 178
Width = 383
Height = 130
Align = alClient
BevelInner = bvLowered
BorderWidth = 4
Caption = 'Panel3'
Caption = 'Panel3' TabOrder = 2

TD (11 1 1 4 4 2
TextHeight = 13
<>object Panel1: TPanel
Left = 0
Top = 0
Width = 383
Height = 41
Align = alTop
TabOrder = 0
<>object DBNavigator:
TDBNavigator
Left = 8
Top = 8
Width = 232
Height = 25
DataSource =
DmEmployee.EmployeeSource
VisibleButtons = [nbFirst,
nbPrior, nbNext, nbLast, nbEdit,
nbPost, nbCancel, nbRefresh]
Ctl3D = False
ParentCtl3D = False
TabOrder = 0
end
<>object BitBtn1: TBitBtn
Left = 315
Top = 8
Width = 60
Height = 25
Hint = 'Exit and close this form'
Caption = 'E&xit'
TabOrder = 1
Kind = bkClose
Style = bsNew
end
end
<>object Panel2: TPanel
Left = 0
Top = 41
Width = 383
Height = 137
Align = alTop
BevelInner = bvLowered
BorderWidth = 4
Caption = 'Panel2'
TabOrder = 1
<>object DBGrid1:
TDBGrid
Left = 6

'PERCENT CHANGE'
Visible = True
end
item
Expanded = False
FieldName = 'UPDATER ID'
Visible = True
end>
end
end
End
Enu
<>object FrmTransDemo: TFrmTransDemo
Left = 244
Top = 150
Width = 506
Height = 342
Hint = 'All edits in this form are
done in a transaction'
ActiveControl = Panel1
Caption = 'Employee Browser' Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11 Font.Name = 'MS Sans Serif'
Font.Style = []
, <u>, , , , , , , , , , , , , , , , , , </u>
OldCreateOrder = True
Position = poScreenCenter
ShowHint = True
OnHide = FormHide
OnShow = FormShow
PixelsPerInch = 96
TextHeight = 13
<>object Panel1: TPanel
Left = 0
Top = 0
Width = 498
Height = 41
Align = alTop
TabOrder = 0
<>object BtnUndoEdits:
TSpeedButton
Left = 345

<>object DBGrid2:
TDBGrid
Left = 6
Top = 6 Width = 371
Height = 118
Hint = 'Salary history is
maintained by a trigger on salary
change'
Align = alClient
BorderStyle = bsNone
DataSource =
DmEmployee.SalaryHistorySource
ReadOnly = True
TabOrder = 0
TitleFont.Charset =
DEFAULT_CHARSET
TitleFont.Color = clBlack
TitleFont.Height = -11
TitleFont.Name = 'MS Sans
Serif'
TitleFont.Style = []
Columns = <
item
Expanded = False
FieldName = 'EMPLOYEE'
Width = 123
Visible = True
end
item
Expanded = False
FieldName =
'CHANGE_DATE'
Visible = True
end
item
Expanded = False
FieldName = 'OLD_SALARY'
Visible = True
end
item
Expanded = False
FieldName = 'NEW SALARY'
Visible = True
end
item
Expanded = False
FieldName =
r iciurvaine –

Align = alClient
BevelInner = bvLowered
BorderWidth = 4
Caption = 'Panel2'
TabOrder = 1
<>object DBGrid1:
TDBGrid
Left = 6
Top = 6
Width = 486
Height = 255
Align = alClient
BorderStyle = bsNone
DataSource =
DmEmployee.EmployeeSource
TabOrder = 0
TitleFont.Charset =
DEFAULT_CHARSET
TitleFont.Color = clBlack
TitleFont.Height = -11
TitleFont.Name = 'MS Sans
Serif'
TitleFont.Style = []
end
end
End
111111111111111111111111111111111111111
<>object FrmQueryProc:
TFrmQueryProc
Left = 279
Top = 170
Width = 381
Height = 367
Hint =
'Explore the Get Emp Proj
procedure in the IBLOCAL alias to
see t' +
'he query procedure'
ActiveControl = Panel1
Caption = 'Employee Project
Assignments'
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
DEFAULI CHARSEI
<u> </u>
Font.Color = clBlack Font.Height = -11

Top = 8
<b>Width</b> = 76
Height = 25
Hint = 'Rolls Back the current
transaction and starts a new one'
Caption = '&Undo Edits'
OnClick = BtnUndoEditsClick
end
<>object BtnCommitEdits:
TSpeedButton
Left = 261
Top = 8
Width = 76
Height = 25
Hint = 'Commits the current
transaction and starts a new one'
Caption = '&Commit Edits'
OnClick = BtnCommitEditsClick
end
<>object DBNavigator:
TDBNavigator
Left = 8
Top = 8
Width = 240
Height = 25
DataSource =
DmEmployee.EmployeeSource
Ctl3D = False
ParentCtl3D = False
TabOrder = 0
end
<>object BitBtn1: TBitBtn
Left = 431
Top = 8
Width = 60
Height = 25
Hint = 'Exit and close this form'
Caption = 'E&xit'
TabOrder = 1
Kind = bkClose
Style = bsNew
end
end
<>object Panel2: TPanel
Left = 0
Top = 41
Width = 498
Height = 267
HUISHU #U!

C 4: ID 101
Caption = 'Panel2'
TabOrder = 1
<>object DBGrid1:
TDBGrid
Left = 6
Top = 6
Width = 361
Height = 125
Hint = 'Select an employee to
execute the query procedure'
Align = alClient
BorderStyle = bsNone
DataSource = EmployeeSource
TabOrder = 0
TitleFont.Charset =
DEFAULT CHARSET
TitleFont.Color = clBlack
TitleFont.Height = -11
TitleFont.Name = 'MS Sans
Serif'
TitleFont.Style = []
end
end
<>object Panel3: TPanel
Left = 0
Top = 178
Width = 373
Height = 136
Align = alClient
BevelInner = bvLowered
BorderWidth = 4
TabOrder = 2
<>object DBGrid2:
TDBGrid
Left = 6
Top = 6
Width = 361
Height = 124
Hint = 'Use SQL Monitor to see
the Get_Emp_Proj query procedure
execute'
Align = alClient
BorderStyle = bsNone
DataSource =
EmployeeProjectsSource
TabOrder = 0
TitleFont.Charset =
DEFAULT_CHARSET

Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
ShowHint = True
OnHide = FormHide
OnShow = FormShow
PixelsPerInch = 96
TextHeight = 13
<>object Panel1: TPanel
Left = 0
Top = 0
Width = 373
Height = 41
Align = alTop
TabOrder = 0
<>object DBNavigator:
TDBNavigator
Left = 8
Top = 8
Width = 240
Height = 25
DataSource =
DmEmployee.EmployeeSource
Ctl3D = False
ParentCtl3D = False
TabOrder = 0
end
<>object BitBtn1: TBitBtn
Left = 305
Top = 8
Width = 60
Height = 25
Hint = 'Exit and close this form'
Caption = 'E&xit'
TabOrder = 1
Kind = bkClose
Style = bsNew
end
end
<>object Panel2: TPanel
Left = 0
Top = 41
Width = 373
Height = 137
Align = alTop
BevelInner = bvLowered
BorderWidth = 4

T 6: 1/0
Left = 168
Top = 72
end
End
\\\\\
<>object FrmLauncher:
TFrmLauncher
Left = 497
Top = 119
BorderIcons = [biSystemMenu,
biMinimize]
BorderStyle = bsSingle
Caption = 'Client/Server Concepts'
ClientHeight = 248
ClientWidth = 240
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
ShowHint = True
PixelsPerInch = 96
TextHeight = 13
<>object BtnTrigg: TButton
Left = 8
Top = 48
Width = 225
Height = 34
Hint = 'Shows a trigger in action'
Caption = 'Salary Change
&Trigger Demo'
TabOrder = 1
OnClick = BtnTriggClick
end
<>object BtnViews: TButton
Left = 8
Top = 8
Width = 225
Height = 34
Hint = 'Demonstrates that views
are treated like tables'
Caption = 'Show a & View in
Suprisi Silvii ii ee i ivii iii

T'd.F4 C.L IDL. I
TitleFont.Color = clBlack
TitleFont.Height = -11
TitleFont.Name = 'MS Sans
Serif'
TitleFont.Style = []
end
end
<>object StatusBar1:
TStatusBar
Left = 0
Top = 314
Width = 373
Height = 19
Panels = <>
SimplePanel = True
end
<>object
EmployeeProjectsQuery: TQuery
DatabaseName =
'EmployeeDemoDB'
SQL.Strings = (
'Select * from Get Emp Proj(
:EMP NO)')
Left = 168
Top = 200
ParamData = <
item
DataType = ftSmallint
Name = 'EMP NO'
ParamType = ptUnknown
Value = 2
end>
end
<>object
•
EmployeeProjectsSource: TDataSource
DataSet =
EmployeeProjectsQuery
Left = 168
Top = 248
end
<>object EmployeeSource:
TDataSource
DataSet =
DmEmployee.EmployeeTable
Enabled = False
OnDataChange =
EmployeeDataChange

OnClick = BtnTransClick	action'
end	TabOrder = 0
End	OnClick = Btn
	end
	<>object B
\\\\\\\\\	Left = 8
	Top = 88
<>object FrmExecProc:	<b>Width = 225</b>
TFrmExecProc	Height = 34
Left = 267	Hint = 'Shows a
Top = 122	in action'
Width = 435	Caption = '&Q
Height = 316	Procedure Demo'
Hint =	TabOrder = 2
'Explore the ShipOrderProc in the	OnClick = Btn0
DmEmployee data model to see wh'	end
+	<>object B
'at shipping an order does'	Left = 8
ActiveControl = Panel1	Top = 128
Caption = 'Sales Review'	<b>Width = 225</b>
Color = clBtnFace	Height = 34
Font.Charset =	Hint = 'Shows a
DEFAULT CHARSET	procedure in action
Font.Color = clBlack	Caption = '&Ex
Font.Height = -11	Procedure Demo'
Font.Name = 'MS Sans Serif'	TabOrder = 3
Font.Style = []	OnClick = Btnl
OldCreateOrder = True	end
Position = poScreenCenter	<>object B
ShowHint = True	Left = 8
OnHide = FormHide	Top = 208
OnShow = FormShow	Width = 225
PixelsPerInch = 96	Height = 34
TextHeight = 13	Hint = 'Exits th
<>object Panel1: TPanel	Caption = '&Ex
Left = 0	TabOrder = 5
Top = 0	OnClick = Btn
Width = 427	end
Height = 41	<>object B
Align = alTop	Left = 8
TabOrder = 0	Top = 168
<>object BtnShipOrder:	$\frac{10p - 103}{\text{Width} = 225}$
TSpeedButton	Height = 34
Left = 258	
Top = 8	Hint = 'Shows s handling'
Width = 64	
	Caption = 'T&1
Height = 25	Demo'
Hint = 'Mark current order as	TabOrder = 4

action'
TabOrder = 0
OnClick = BtnViewsClick
end
<>object BtnQrySP: TButton
Left = 8
Top = 88
Width = 225
Height = 34
Hint = 'Shows a query procedure
in action'
Caption = '&Query Stored
Procedure Demo'
TabOrder = 2
OnClick = BtnQrySPClick
end
<>object BtnExecSP: TButton
Left = 8
Top = 128
Width = 225
Height = 34
Hint = 'Shows an executable
procedure in action'
Caption = '&Executable Stored
Procedure Demo'
TabOrder = 3
OnClick = BtnExecSPClick
end
<>object BtnClose: TButton Left = 8
Top = 208
Width = 225
Height = 34
Hint = 'Exits this sample'
Caption = '&Exit'
TabOrder = 5
OnClick = BtnCloseClick
end
<>object BtnTrans: TButton
Left = 8
Top = 168
Width = 225
Height = 34
Hint = 'Shows simple transaction
handling'
Caption = 'T&ransaction Editing
Demo'
TabOrder = 4

DandanStyla – bsNana
BorderStyle = bsNone DataSource =
DmEmployee.SalesSource
TabOrder = 0
TitleFont.Charset =
DEFAULT_CHARSET
TitleFont.Color = clBlack
TitleFont.Height = -11
TitleFont.Name = 'MS Sans
Serif'
TitleFont.Style = []
end
end
<>object Panel3: TPanel
Left = 0
Top = 172
Width = 427
Height = 110
Hint = 'Customers are linked to
sales in the data model'
Align = alBottom
BevelInner = bvLowered
BorderWidth = 4
Caption = 'Panel3'
TabOrder = 2
<>object ScrollBox:
TScrollBox
Left = 6
Top = 6
Width = 415
Height = 98
HorzScrollBar.Margin = 6
VertScrollBar.Margin = 6
Align = alClient
BorderStyle = bsNone
TabOrder = 0
<>object Label1: TLabel
Left = 209
Top = 6
Width = 64
Height = 13
Alignment = taRightJustify
Caption = 'Customer No:'
FocusControl = EditCUST_NO
end
<>object Label2: TLabel
Left = 1
<b>Top</b> = 7

shipped'
Caption = '&Ship Order'
Enabled = False
OnClick = BtnShipOrderClick
end
<>object DBNavigator:
TDBNavigator
Left = 8
Top = 8
Width = 240
Height = 25
DataSource =
DmEmployee.SalesSource
Ctl3D = False
ParentCtl3D = False
TabOrder = 0
end
<>object BitBtn1: TBitBtn
Left = 359
Top = 8
Width = 60
Height = 25
Hint = 'Exit and close this form'
Caption = 'E&xit'
TabOrder = 1
Kind = bkClose
Style = bsNew
end
end
<>object Panel2: TPanel
Left = 0
<b>Top</b> = 41
Width = 427
Height = 131
Align = alClient
BevelInner = bvLowered
BorderWidth = 4
Caption = 'Panel2'
TabOrder = 1
<>object DBGrid1:
TDBGrid
Left = 6
Top = 6
Width = 415
Height = 119
Hint = 'Select an open order to
ship the order'
Align = alClient

Left = 51
Top = 4
Width = 125
Height = 21
DataField = 'CUSTOMER'
DataSource =
DmEmployee.CustomerSource
TabOrder = 1
end
<>object
EditCONTACT_FIRST: TDBEdit
Left = 51
Top = 26
Width = 75
Height = 21
DataField =
'CONTACT_FIRST'
DataSource =
DmEmployee.CustomerSource
TabOrder = 2
end
<>object
EditCONTACT_LAST: TDBEdit
Left = 129
Top = 26
Width = 100
Height = 21
DataField =
'CONTACT LAST'
DataSource =
<b>DmEmployee.CustomerSource</b>
TabOrder = 3
end
<>object EditPHONE_NO:
TDBEdit
Left = 276
Top = 26
Width = 120
Height = 21
DataField = 'PHONE NO'
DataField = THONE_NO  DataSource =
DmEmployee.CustomerSource
TabOrder = 4
end
<>object
EditADDRESS LINE: TDBEdit
Left = 51
Top = 48

<b>Width</b> = 47
Height = 13
Alignment = taRightJustify
Caption = 'Company:'
FocusControl =
EditCUSTOMER
end
<>object Label3: TLabel
Left = 8
Top = 29
Width = 40
Height = 13
Alignment = taRightJustify
Caption = 'Contact:'
FocusControl =
EditCONTACT FIRST
end
<>object Label6: TLabel Left = 7
Top = 50
Width = 41
Height = 13
Alignment = taRightJustify
Caption = 'Address:'
FocusControl =
EditADDRESS_LINE
end
<>object Label4: TLabel
Left = 239
Top = 29
<b>Width</b> = 34
Height = 13
Alignment = taRightJustify
Caption = 'Phone:'
end
<>object EditCUST NO:
TDBEdit
Left = 276
Top = 3
Width = 49
Height = 21
DataField = 'CUST NO'
DataField = CUS1_NO  DataSource =
DmEmployee.CustomerSource
TabOrder = 0
end
<pre>&lt;&gt;object EditCUSTOMER:</pre>
TDBEdit

Height = 21  DataField = 'COUNTRY'  DataSource =  DmEmployee.CustomerSource  TabOrder = 9  end  <>object  EditPOSTAL_CODE: TDBEdit  Left = 336  Top = 70  Width = 60  Height = 21  DataField = 'POSTAL_CODE'  DataSource =  DmEmployee.CustomerSource  TabOrder = 10  end  <>object DBCheckBox1:  TDBCheckBox  Left = 333  Top = 3  Width = 67  Height = 17  Caption = 'On Hold'  DataField = 'ON_HOLD'  DataSource =  DmEmployee.CustomerSource  Font.Charset =  DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = [fsBold]  ParentFont = False  TabOrder = 11  ValueChecked = 'True;*;Yes'  ValueUnchecked = 'False; ;;NIL;No'  end end end end end end Source  DataSet =  DmEmployee.SalesSource:  TDataSource  DataSource  DataSet =  DmEmployee.SalesTable  Enabled = False  OnDataChange =  SalesSourceDataChange	
DataSource = DmEmployee.CustomerSource TabOrder = 9 end <>object EditPOSTAL_CODE: TDBEdit Left = 336 Top = 70 Width = 60 Height = 21 DataField = 'POSTAL_CODE' DataSource = DmEmployee.CustomerSource TabOrder = 10 end <>object DBCheckBox1: TDBCheckBox Left = 333 Top = 3 Width = 67 Height = 17 Caption = 'On Hold' DataField = 'ON_HOLD' DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end	Height = 21
DmEmployee.CustomerSource TabOrder = 9 end <>object EditPOSTAL_CODE: TDBEdit Left = 336 Top = 70 Width = 60 Height = 21 DataField = 'POSTAL_CODE' DataSource = DmEmployee.CustomerSource TabOrder = 10 end <>object DBCheckBox1: TDBCheckBox Left = 333 Top = 3 Width = 67 Height = 17 Caption = 'On Hold' DataField = 'ON_HOLD' DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False;;;NIL;No' end	
TabOrder = 9 end  <>object EditPOSTAL_CODE: TDBEdit Left = 336 Top = 70 Width = 60 Height = 21 DataField = 'POSTAL_CODE' DataSource = DmEmployee.CustomerSource TabOrder = 10 end <>object DBCheckBox1: TDBCheckBox Left = 333 Top = 3 Width = 67 Height = 17 Caption = 'On Hold' DataField = 'ON_HOLD' DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False;;;NIL;No' end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
end <>object EditPOSTAL_CODE: TDBEdit Left = 336 Top = 70 Width = 60 Height = 21 DataField = 'POSTAL_CODE' DataSource = DmEmployee.CustomerSource TabOrder = 10 end <>object DBCheckBox1: TDBCheckBox Left = 333 Top = 3 Width = 67 Height = 17 Caption = 'On Hold' DataField = 'ON_HOLD' DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False;;;NIL;No' end end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
<pre>&lt;&gt;object EditPOSTAL_CODE: TDBEdit    Left = 336    Top = 70    Width = 60    Height = 21    DataField = 'POSTAL_CODE'    DataSource = DmEmployee.CustomerSource    TabOrder = 10    end    &lt;&gt;object DBCheckBox1: TDBCheckBox    Left = 333    Top = 3    Width = 67    Height = 17    Caption = 'On Hold'    DataField = 'ON_HOLD'    DataSource = DmEmployee.CustomerSource    Font.Charset = DEFAULT_CHARSET    Font.Color = clBlack    Font.Height = -11    Font.Name = 'MS Sans Serif'    Font.Style = [fsBold]    ParentFont = False    TabOrder = 11    ValueChecked = 'True;*;Yes'    ValueUnchecked = 'False;;;NIL;No'    end    end    end    end end c&gt;object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =</pre>	TabOrder = 9
EditPOSTAL_CODE: TDBEdit  Left = 336  Top = 70  Width = 60  Height = 21  DataField = 'POSTAL_CODE'  DataSource =  DmEmployee.CustomerSource  TabOrder = 10  end  <>object DBCheckBox1:  TDBCheckBox  Left = 333  Top = 3  Width = 67  Height = 17  Caption = 'On Hold'  DataField = 'ON_HOLD'  DataSource =  DmEmployee.CustomerSource  Font.Charset =  DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = [fsBold]  ParentFont = False  TabOrder = 11  ValueChecked = 'True;*;Yes'  ValueUnchecked = 'False; ;;NIL;No'  end end  end  <>object SalesSource:  TDataSource  DataSet =  DmEmployee.SalesTable  Enabled = False  OnDataChange =	end
Left = 336 Top = 70 Width = 60 Height = 21 DataField = 'POSTAL_CODE' DataSource = DmEmployee.CustomerSource TabOrder = 10 end <>object DBCheckBox1: TDBCheckBox Left = 333 Top = 3 Width = 67 Height = 17 Caption = 'On Hold' DataField = 'ON_HOLD' DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
Top = 70 Width = 60 Height = 21 DataField = 'POSTAL_CODE' DataSource = DmEmployee.CustomerSource TabOrder = 10 end <>object DBCheckBox1: TDBCheckBox Left = 333 Top = 3 Width = 67 Height = 17 Caption = 'On Hold' DataField = 'ON_HOLD' DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	EditPOSTAL_CODE: TDBEdit
Width = 60 Height = 21 DataField = 'POSTAL_CODE' DataSource = DmEmployee.CustomerSource TabOrder = 10 end <>object DBCheckBox1: TDBCheckBox Left = 333 Top = 3 Width = 67 Height = 17 Caption = 'On Hold' DataField = 'ON_HOLD' DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	Left = 336
Height = 21 DataField = 'POSTAL_CODE' DataSource = DmEmployee.CustomerSource TabOrder = 10 end <>object DBCheckBox1: TDBCheckBox Left = 333 Top = 3 Width = 67 Height = 17 Caption = 'On Hold' DataField = 'ON_HOLD' DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	Top = 70
DataField = 'POSTAL CODE' DataSource = DmEmployee.CustomerSource TabOrder = 10 end <>object DBCheckBox1: TDBCheckBox Left = 333 Top = 3 Width = 67 Height = 17 Caption = 'On Hold' DataField = 'ON HOLD' DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	Width = 60
DataSource = DmEmployee.CustomerSource TabOrder = 10 end <>object DBCheckBox1: TDBCheckBox Left = 333 Top = 3 Width = 67 Height = 17 Caption = 'On Hold' DataField = 'ON_HOLD' DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	Height = 21
DataSource = DmEmployee.CustomerSource TabOrder = 10 end <>object DBCheckBox1: TDBCheckBox Left = 333 Top = 3 Width = 67 Height = 17 Caption = 'On Hold' DataField = 'ON_HOLD' DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	DataField = 'POSTAL CODE'
TabOrder = 10 end <>object DBCheckBox1: TDBCheckBox Left = 333 Top = 3 Width = 67 Height = 17 Caption = 'On Hold' DataField = 'ON_HOLD' DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
TabOrder = 10 end <>object DBCheckBox1: TDBCheckBox Left = 333 Top = 3 Width = 67 Height = 17 Caption = 'On Hold' DataField = 'ON_HOLD' DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	DmEmployee.CustomerSource
end <>object DBCheckBox1: TDBCheckBox  Left = 333  Top = 3  Width = 67  Height = 17  Caption = 'On Hold'  DataField = 'ON_HOLD'  DataSource =  DmEmployee.CustomerSource  Font.Charset =  DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = [fsBold]  ParentFont = False  TabOrder = 11  ValueChecked = 'True;*;Yes'  ValueUnchecked = 'False; ;;NIL;No'  end end end  end  c>object SalesSource: TDataSource  DataSet =  DmEmployee.SalesTable  Enabled = False  OnDataChange =	
<pre>&lt;&gt;object DBCheckBox1: TDBCheckBox     Left = 333     Top = 3     Width = 67     Height = 17     Caption = 'On Hold'     DataField = 'ON_HOLD'     DataSource =     DmEmployee.CustomerSource     Font.Charset = DEFAULT_CHARSET     Font.Color = clBlack     Font.Height = -11     Font.Name = 'MS Sans Serif'     Font.Style = [fsBold]     ParentFont = False     TabOrder = 11     ValueChecked = 'True;*;Yes'     ValueUnchecked = 'False; ;;NIL;No'     end     end end &lt;&gt;object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =</pre>	
TDBCheckBox  Left = 333  Top = 3  Width = 67  Height = 17  Caption = 'On Hold'  DataField = 'ON_HOLD'  DataSource =  DmEmployee.CustomerSource  Font.Charset =  DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = [fsBold]  ParentFont = False  TabOrder = 11  ValueChecked = 'True;*;Yes'  ValueUnchecked = 'False; ;;NIL;No'  end end end end  C>object SalesSource:  TDataSource  DataSet =  DmEmployee.SalesTable  Enabled = False  OnDataChange =	
Left = 333 Top = 3 Width = 67 Height = 17 Caption = 'On Hold' DataField = 'ON_HOLD' DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False;;;NIL;No' end end end end C>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
Top = 3 Width = 67 Height = 17 Caption = 'On Hold' DataField = 'ON_HOLD' DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end end end End Enabled = False OnDataChange =	
Width = 67 Height = 17 Caption = 'On Hold' DataField = 'ON_HOLD' DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end e	
Height = 17 Caption = 'On Hold' DataField = 'ON_HOLD' DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False;;;NIL;No' end end end end C>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
Caption = 'On Hold' DataField = 'ON_HOLD' DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False;;;NIL;No' end end end end C>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
DataField = 'ON_HOLD' DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False;;;NIL;No' end end end end C>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
DataSource = DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end end C>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	DataField = 'ON HOLD'
DmEmployee.CustomerSource Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
Font.Charset = DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = [fsBold]  ParentFont = False  TabOrder = 11  ValueChecked = 'True;*;Yes'  ValueUnchecked = 'False; ;;NIL;No' end end end  end  C>object SalesSource:  TDataSource  DataSet = DmEmployee.SalesTable  Enabled = False  OnDataChange =	
DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end end C>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	_
Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
Font.Style = [fsBold] ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
ParentFont = False TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
TabOrder = 11 ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
ValueChecked = 'True;*;Yes' ValueUnchecked = 'False; ;;NIL;No' end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
ValueUnchecked = 'False; ;;NIL;No' end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
;;NIL;No' end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	, ,
end end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	<i>′</i>
end end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
end <>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
<>object SalesSource: TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
TDataSource DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	
DataSet = DmEmployee.SalesTable Enabled = False OnDataChange =	<u> </u>
DmEmployee.SalesTable Enabled = False OnDataChange =	
Enabled = False OnDataChange =	DataSet =
OnDataChange =	DmEmployee.SalesTable
S	Enabled = False
SalesSourceDataChange	OnDataChange =
<u> </u>	SalesSourceDataChange

Width = 171
Height = 21
DataField =
'ADDRESS_LINE1'
DataSource =
DmEmployee.CustomerSource
TabOrder = 5
end
<>object
EditADDRESS_LINE2: TDBEdit
Left = 226
Top = 48
Width = 170
Height = 21
DataField =
'ADDRESS LINE2'
DataSource =
DmEmployee.CustomerSource
TabOrder = 6
end
<>object EditCITY:
TDBEdit
Left = 51
Top = 70
Width = 125
Height = 21
DataField = 'CITY'
DataField - C111  DataSource =
DmEmployee.CustomerSource TabOrder = 7
end
<>object
EditSTATE_PROVINCE: TDBEdit
Left = 180
Top = 70
Width = 75
Height = 21
DataField =
'STATE_PROVINCE'
DataSource =
DmEmployee.CustomerSource
TabOrder = 8
end
<>object EditCOUNTRY:
TDBEdit
Left = 258
Top = 70
Width = 75

Required = True
end
<>object
SalesTableSALES_REP:
<b>TSmallintField</b>
DisplayLabel = 'Sales Rep'
FieldName = 'SALES_REP'
end
<>object
SalesTableORDER STATUS:
TStringField
DisplayLabel = 'Order Status'
FieldName = 'ORDER_STATUS'
Required = True
Size = 7
end
<>object
SalesTableORDER_DATE:
<b>TDateTimeField</b>
DisplayLabel = 'Order Date'
FieldName = 'ORDER_DATE'
Required = True
end
<>object
SalesTableSHIP DATE:
TDateTimeField
DisplayLabel = 'Ship Date' FieldName = 'SHIP DATE'
end
<>object
SalesTableDATE_NEEDED:
<b>TDateTimeField</b>
DisplayLabel = 'Date Needed'
FieldName = 'DATE_NEEDED'
end
<>object SalesTablePAID:
<b>TStringField</b>
DisplayLabel = 'Paid'
FieldName = 'PAID'
Size = 1
end
<>object
<u>~</u>
SalesTableQTY_ORDERED:
TIntegerField
DisplayLabel = 'Quantity
Ordered'
FieldName = 'QTY_ORDERED'
Required = True

1 -64 200
Left = 200
Top = 80
end
End
\\\
<>object DmEmployee:
TDmEmployee
OldCreateOrder = True
OnCreate = DmEmployeeCreate
Left = 126
Top = 130
Height = 188
Width = 414
<>object EmployeeDatabase:
TDatabase
AliasName = 'IBLOCAL'
Connected = True
DatabaseName =
'EmployeeDemoDB'
LoginPrompt = False
Params.Strings = (
'USER NAME=SYSDBA'
'PASSWORD=masterkey')
SessionName = 'Default'
Left = 40
Top = 8
end
<>object SalesTable: TTable
DatabaseName =
'EmployeeDemoDB'
IndexFieldNames =
'PO_NUMBER'
TableName = 'SALES'
Left = 40
Top = 56
<>object
SalesTablePO_NUMBER:
TStringField
DisplayLabel = 'PO Number'
FieldName = 'PO_NUMBER'
Required = True
Size = 8
end
<>object
SalesTableCUST_NO: TIntegerField
DisplayLabel = 'Customer No'
FieldName = 'CUST_NO'

FieldName = 'CUSTOMER' Required = True size = 25 end <>object CustomerTableCONTACT_FIRST: TStringField DisplayLabel = 'Contact1' FieldName = 'CONTACT_FIRST' Size = 15 end CustomerTableCONTACT_LAST: Size = 15 End CustomerTableCONTACT_LAST:  FieldName = 'ITEM_TYPE' Size = 12		
<pre> &lt;&gt;object CustomerTableCUSTOMER: TStringField DisplayLabel = 'Customer' FieldName = 'CUSTOMER' Required = True Size = 25 end &lt;&gt;object CustomerTableCONTACT_FIRST: TStringField DisplayLabel = 'Contact1' FieldName = 'CONTACT_FIRST' Size = 15 end  &lt;&gt;object CustomerTableCONTACT_LAST: Size = 15  End  SalesTableTOTAL_VALUE: TIntegerField DisplayLabel = 'Total Value' FieldName = 'TOTAL_VALUE Required = True end &gt;object SalesTableDISCOUNT: TFloatFiel DisplayLabel = 'Discount' Required = True end&gt;object SalesTableITEM_TYPE: TStringField DisplayLabel = 'Item Type' FieldName = 'ITEM_TYPE' Size = 12</pre>	Required = True	end
<pre> &lt;&gt;object CustomerTableCUSTOMER: TStringField DisplayLabel = 'Customer' FieldName = 'CUSTOMER' Required = True Size = 25 end &lt;&gt;object CustomerTableCONTACT_FIRST: TStringField DisplayLabel = 'Contact1' FieldName = 'CONTACT_FIRST' Size = 15 end  &lt;&gt;object CustomerTableCONTACT_LAST: Size = 15  End  SalesTableTOTAL_VALUE: TIntegerField DisplayLabel = 'Total Value' FieldName = 'TOTAL_VALUE Required = True end &gt;object SalesTableDISCOUNT: TFloatFiel DisplayLabel = 'Discount' Required = True end&gt;object SalesTableITEM_TYPE: TStringField DisplayLabel = 'Item Type' FieldName = 'ITEM_TYPE' Size = 12</pre>	end	<>object
TStringField  DisplayLabel = 'Customer' FieldName = 'CUSTOMER' Required = True Size = 25 end <>object CustomerTableCONTACT_FIRST: TStringField DisplayLabel = 'Contact1' FieldName = 'CONTACT_FIRST' Size = 15 end DisplayLabel = 'ITOTAL_VALUE Required = True end <>object SalesTableDISCOUNT: TFloatFiel DisplayLabel = 'Discount' Required = True end>object SalesTableITEM_TYPE: TStringField DisplayLabel = 'Item Type' FieldName = 'ITEM_TYPE' Size = 12	<>object	
DisplayLabel = 'Customer' FieldName = 'CUSTOMER' Required = True Size = 25 end <>object CustomerTableCONTACT_FIRST: TStringField DisplayLabel = 'Contact1' FieldName = 'CONTACT_FIRST' Size = 15 end CustomerTableCONTACT_LAST: FieldName = 'TOTAL_VALUE Required = True end <>object SalesTableDISCOUNT: TFloatFiel DisplayLabel = 'Discount' Required = True end>object SalesTableITEM_TYPE: TStringField DisplayLabel = 'Item Type' FieldName = 'ITEM TYPE' Size = 12	CustomerTableCUSTOMER:	TIntegerField
DisplayLabel = 'Customer' FieldName = 'CUSTOMER' Required = True Size = 25 end <>object CustomerTableCONTACT_FIRST: TStringField DisplayLabel = 'Contact1' FieldName = 'CONTACT_FIRST' Size = 15 end CustomerTableCONTACT_LAST: FieldName = 'TOTAL_VALUE Required = True end <>object SalesTableDISCOUNT: TFloatFiel DisplayLabel = 'Discount' Required = True end>object SalesTableITEM_TYPE: TStringField DisplayLabel = 'Item Type' FieldName = 'ITEM TYPE' Size = 12	TStringField	DisplayLabel = 'Total Value'
FieldName = 'CUSTOMER'  Required = True  Size = 25 end  <>object CustomerTableCONTACT_FIRST: TStringField DisplayLabel = 'Contact1' FieldName = 'CONTACT_FIRST' Size = 15 end >object CustomerTableCONTACT_LAST:  FieldName = 'True end>object SalesTableDISCOUNT: TFloatFiel DisplayLabel = 'DISCOUNT' Required = True end>object SalesTableITEM_TYPE: TStringField DisplayLabel = 'Item Type' FieldName = 'ITEM_TYPE' Size = 12	DisplayLabel = 'Customer'	FieldName = 'TOTAL VALUE'
Required = True Size = 25 end  <>object CustomerTableCONTACT_FIRST: TStringField DisplayLabel = 'Contact1' FieldName = 'CONTACT_FIRST' Size = 15 end >object SalesTableDISCOUNT: TFloatFiel DisplayLabel = 'DISCOUNT' Required = True end>object SalesTableITEM_TYPE: TStringField DisplayLabel = 'Item Type' FieldName = 'ITEM_TYPE' Size = 12		Required = True
Size = 25 end  <>object CustomerTableCONTACT_FIRST: TStringField DisplayLabel = 'Contact1' FieldName = 'CONTACT_FIRST' Size = 15 end >object SalesTableDISCOUNT: TFloatFiel DisplayLabel = 'DISCOUNT' Required = True end>object SalesTableITEM_TYPE: TStringField DisplayLabel = 'Item Type' FieldName = 'ITEM_TYPE' Size = 12		-
end  <>object CustomerTableCONTACT_FIRST: TStringField DisplayLabel = 'Contact1' FieldName = 'CONTACT_FIRST' Size = 15 end >object CustomerTableCONTACT_LAST:  SalesTableDISCOUNT: TFloatFiel DisplayLabel = 'Discount' Required = True end>object SalesTableITEM_TYPE: TStringField DisplayLabel = 'Item Type' FieldName = 'ITEM_TYPE' Size = 12	-	<>object
<pre> &lt;&gt;object CustomerTableCONTACT_FIRST: TStringField DisplayLabel = 'Contact1' FieldName = 'CONTACT_FIRST' Size = 15 end  &lt;&gt;object CustomerTableCONTACT_LAST:  DisplayLabel = 'Discount' FieldName = 'DISCOUNT' Required = True end &lt;&gt;object SalesTableITEM_TYPE: TStringField DisplayLabel = 'Item Type' FieldName = 'ITEM TYPE' Size = 12</pre>	end	
CustomerTableCONTACT_FIRST: TStringField  DisplayLabel = 'Contact1'  FieldName = 'DISCOUNT'  Required = True  end  <>object  SalesTableITEM_TYPE: TStringField  DisplayLabel = 'Item Type'  FieldName = 'ITEM_TYPE'  Size = 12	<>object	
TStringField  DisplayLabel = 'Contact1'  FieldName = 'CONTACT_FIRST'  Size = 15 end  end  CustomerTableCONTACT_LAST:  Required = True end  SalesTableITEM_TYPE: TStringField DisplayLabel = 'Item Type' FieldName = 'ITEM_TYPE' Size = 12	•	
DisplayLabel = 'Contact1'  FieldName = 'CONTACT_FIRST'  Size = 15  end  end  CustomerTableCONTACT_LAST:  end  customerTableContact1'  end  customerTableContact1'  end  customerTableContact1'  end  customerTableITEM_TYPE:  TStringField  DisplayLabel = 'Item Type'  FieldName = 'ITEM_TYPE'  Size = 12	<del>-</del>	
FieldName =		-
'CONTACT_FIRST'SalesTableITEM_TYPE:Size = 15TStringFieldendDisplayLabel = 'Item Type'<>objectFieldName = 'ITEM_TYPE'CustomerTableCONTACT_LAST:Size = 12	1 0	
Size = 15 end CustomerTableCONTACT_LAST:  TStringField DisplayLabel = 'Item Type' FieldName = 'ITEM_TYPE' Size = 12		•
end >object CustomerTableCONTACT_LAST:  DisplayLabel = 'Item Type' FieldName = 'ITEM_TYPE' Size = 12	_	<u>—</u>
<pre>&lt;&gt;object CustomerTableCONTACT_LAST:</pre> FieldName = 'ITEM_TYPE' Size = 12		
CustomerTableCONTACT_LAST: Size = 12		
_	<u> </u>	
	TStringField	end
		<>object SalesTableAGED:
FieldName = TFloatField		•
'CONTACT_LAST'  DisplayLabel = 'Aged'		
end FieldName = 'AGED'		
<>object end		
CustomerTablePHONE_NO: end		
TStringField <>object SalesSource:		
DisplayLabel = 'Phone No'  TDataSource	DisplayLabel = 'Phone No'	· · · · · · · · · · · · · · · · · · ·
FieldName = 'PHONE_NO' DataSet = SalesTable		
end Left = 40	end	
<>object	<>object	
CustomerTableADDRESS_LINE1: end	CustomerTableADDRESS_LINE1:	-
TStringField <>object CustomerTable:	TStringField	<>object CustomerTable:
DisplayLabel = 'Address 1' TTable	DisplayLabel = 'Address 1'	<u> </u>
FieldName = DatabaseName =	FieldName =	DatabaseName =
'ADDRESS_LINE1' 'EmployeeDemoDB'	'ADDRESS_LINE1'	'EmployeeDemoDB'
Size = 30 IndexFieldNames = 'CUST_NO'	Size = 30	IndexFieldNames = 'CUST_NO'
end MasterFields = 'CUST_NO'	end	MasterFields = 'CUST_NO'
<>object MasterSource = SalesSource	1	MasterSource = SalesSource
CustomerTableADDRESS_LINE2: TableName = 'CUSTOMER'	_	TableName = 'CUSTOMER'
TStringField Left = 128		Left = 128
DisplayLabel = 'Address 2'  Top = 56		Top = 56
FieldName = <>object		<>object
'ADDRESS_LINE2' CustomerTableCUST NO:		CustomerTableCUST NO:
Size = 30 TIntegerField		
end DisplayLabel = 'Customer No'	end	DisplayLabel = 'Customer No'
<>object FieldName = 'CUST NO'		
CustomerTableCITY: TStringField	1	

!EmployeeDeme DD!	Display Label - !City!
'EmployeeDemoDB'	DisplayLabel = 'City'
IndexFieldNames = 'EMP_NO'	FieldName = 'CITY'
TableName = 'EMPLOYEE'	Size = 25
Left = 224	end
Top = 56	<>object
<>object	CustomerTableSTATE_PROVINCE
EmployeeTableEMP_NO:	: TStringField
TSmallintField	DisplayLabel = 'State or
DisplayLabel = 'Employee	Province'
Number'	FieldName =
FieldName = 'EMP_NO'	'STATE_PROVINCE'
Required = True	Size = 15
end	end
<>object	<>object
EmployeeTableFIRST_NAME:	CustomerTableCOUNTRY:
TStringField	TStringField
DisplayLabel = 'First Name'	DisplayLabel = 'Country'
FieldName = 'FIRST_NAME'	FieldName = 'COUNTRY'
Required = True	Size = 15
Size = 15	end
end	<>object
<>object	CustomerTablePOSTAL_CODE:
EmployeeTableLAST_NAME:	TStringField
TStringField	DisplayLabel = 'Postal Code'
DisplayLabel = 'Last Name'	FieldName = 'POSTAL_CODE'
FieldName = 'LAST_NAME'	Size = 12
Required = True	end
end	<>object
<>object	CustomerTableON_HOLD:
EmployeeTablePHONE_EXT:	TStringField
TStringField	DisplayLabel = 'On Hold'
DisplayLabel = 'Phone	FieldName = 'ON_HOLD'
Extension'	Size = 1
FieldName = 'PHONE_EXT'	end
Size = 4	end
end	<>object CustomerSource:
<>object	<b>TDataSource</b>
EmployeeTableHIRE_DATE:	DataSet = CustomerTable
TDateTimeField	Left = 128
DisplayLabel = 'Hire Date'	Top = 104
FieldName = 'HIRE_DATE'	end
Required = True	<>object EmployeeTable:
end	TTable
<>object	AfterPost =
EmployeeTableDEPT NO:	<b>EmployeeTableAfterPost</b>
TStringField	BeforeDelete =
DisplayLabel = 'Dept No'	<b>EmployeeTableBeforeDelete</b>
FieldName = 'DEPT NO'	DatabaseName =
<u> </u>	

TTable	Dogwined - Two
	Required = True Size = 3
DatabaseName =	
'EmployeeDemoDB'	end
IndexFieldNames = 'EMP_NO'	<>object
MasterFields = 'EMP_NO'	EmployeeTableJOB_CODE:
MasterSource = EmployeeSource	TStringField
TableName =	DisplayLabel = 'Job Code'
'SALARY_HISTORY'	FieldName = 'JOB_CODE'
Left = 328	Required = True
Top = 56	Size = 5
<>object	end
SalaryHistoryTableEMPLOYEE:	<>object
TStringField	EmployeeTableJOB_GRADE:
DisplayLabel = 'Employee'	<b>TSmallintField</b>
FieldKind = fkLookup	DisplayLabel = 'Job Grade'
FieldName = 'EMPLOYEE'	FieldName = 'JOB_GRADE'
LookupDataSet =	Required = True
EmployeeLookup	end
LookupKeyFields = 'EMP_NO'	<>object
LookupResultField =	EmployeeTableJOB COUNTRY:
'FULL NAME'	TStringField
KeyFields = 'EMP NO'	DisplayLabel = 'Job Country'
Size = 37	FieldName = 'JOB COUNTRY'
Lookup = True	Required = True
end	Size = 15
<>object	end
SalaryHistoryTableEMP NO:	<>object
TSmallintField	EmployeeTableSALARY:
FieldName = 'EMP NO'	TFloatField
Required = True	DisplayLabel = 'Salary'
DisplayFormat = 'Employee	FieldName = 'SALARY'
Number'	Required = True
end	end
<>object	<>object
SalaryHistoryTableCHANGE DAT	EmployeeTableFULL NAME:
E: TDateTimeField	TStringField
DisplayLabel = 'Change Date'	DisplayLabel = 'Name'
FieldName = 'CHANGE DATE'	FieldName = 'FULL NAME'
Required = True	Size = 37
end	end
<>object	end
SalaryHistoryTableUPDATER ID:	<>object EmployeeSource:
TStringField	TDataSource
DisplayLabel = 'Updater ID'	DataSource  DataSet = EmployeeTable
FieldName = 'UPDATER ID'	Left = 224
Required = True	
_ •	Top = 104
end	end
<>object	<>object SalaryHistoryTable:

Left = 224
Top = 8
ParamData = <
item
DataType = ftInteger
Name = 'EMP_NUM'
ParamType = ptInput
end>
end
<>object EmployeeLookup:
TTable
AfterPost =
EmployeeTableAfterPost
BeforeDelete =
EmployeeTableBeforeDelete
DatabaseName =
'EmployeeDemoDB'
IndexFieldNames = 'EMP NO'
TableName = 'EMPLOYEE'
Left = 328
Top = 8
<>object SmallintField1:
TSmallintField
DisplayLabel = 'Employee
Number'
FieldName = 'EMP NO'
Required = True
end
<>object StringField1:
TStringField
DisplayLabel = 'First Name'
FieldName = 'FIRST NAME'
Required = True
Size = 15
end
<>object StringField2:
TStringField
DisplayLabel = 'Last Name'
FieldName = 'LAST_NAME'
Required = True
end
<>object StringField3:
TStringField DisplayLabel = 'Phone
DisplayLabel = 'Phone Extension'
FieldName = 'PHONE EXT'
Size = 4
end

SalaryHistoryTableOLD_SALARY:
TFloatField
DisplayLabel = 'Old Salary'
FieldName = 'OLD_SALARY'
Required = True
end
<>object
SalaryHistoryTablePERCENT CH
ANGE: TFloatField
DisplayLabel = '% Change'
FieldName =
'PERCENT CHANGE'
Required = True
end
<>object
SalaryHistoryTableNEW SALARY:
TFloatField
DisplayLabel = 'New Salary'
FieldName = 'NEW SALARY'
end
end
<>object
SalaryHistorySource: TDataSource
DataSet = SalaryHistoryTable
Left = 328
Top = 104
end
<>object ShipOrderProc:
TStoredProc
DatabaseName =
'EmployeeDemoDB'
StoredProcName =
'SHIP_ORDER'
Left = 128
Top = 8
ParamData = <
item
DataType = ftString
Name = 'PO NUM'
ParamType = ptInput
end>
end
<>object
DeleteEmployeeProc: TStoredProc
DatabaseName =
'EmployeeDemoDB'
StoredProcName =
'DELETE_EMPLOYEE'

111111111111111111111111111111111111111
-----------------------------------------

<pre>&lt;&gt;object DateTimeField1: TDateTimeField     DisplayLabel = 'Hire Date'     FieldName = 'HIRE_DATE'     Required = True     end     &lt;&gt;object StringField4: TStringField     DisplayLabel = 'Dept No'     FieldName = 'DEPT_NO'     Required = True     Size = 3     end     &lt;&gt;object StringField5: TStringField     DisplayLabel = 'Job Code'     FieldName = 'JOB_CODE'     Required = True     Size = 5     end     &lt;&gt;object SmallintField2: TSmallintField     DisplayLabel = 'Job Grade'     FieldName = 'JOB_GRADE'     Required = True     end     &lt;&gt;object StringField6: TStringField     DisplayLabel = 'Job Country'     FieldName = 'JOB_COUNTRY'     Required = True     Size = 15     end     &lt;&gt;object FloatField1: TFloatField</pre>
DisplayLabel = 'Hire Date' FieldName = 'HIRE_DATE' Required = True end <>object StringField4: TStringField DisplayLabel = 'Dept No' FieldName = 'DEPT_NO' Required = True Size = 3 end <>object StringField5: TStringField DisplayLabel = 'Job Code' FieldName = 'JOB_CODE' Required = True Size = 5 end <>object SmallintField2: TSmallintField DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE' Required = True end <>object StringField6: TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end <>object FloatField1: TFloatField
FieldName = 'HIRE_DATE'  Required = True end  <>object StringField4: TStringField  DisplayLabel = 'Dept No' FieldName = 'DEPT_NO'  Required = True Size = 3 end  <>object StringField5: TStringField  DisplayLabel = 'Job Code' FieldName = 'JOB_CODE'  Required = True Size = 5 end  <>object SmallintField2: TSmallintField  DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE'  Required = True end  <>object StringField6: TStringField  DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end  <>object FloatField1: TFloatField
Required = True end  <>object StringField4: TStringField  DisplayLabel = 'Dept No' FieldName = 'DEPT_NO' Required = True Size = 3 end  <>object StringField5: TStringField  DisplayLabel = 'Job Code' FieldName = 'JOB_CODE' Required = True Size = 5 end  <>object SmallintField2: TSmallintField  DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE' Required = True end  <>object StringField6: TStringField  DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end  <>object FloatField1: TFloatField
end  <>object StringField4: TStringField  DisplayLabel = 'Dept No' FieldName = 'DEPT_NO' Required = True Size = 3 end  <>object StringField5: TStringField  DisplayLabel = 'Job Code' FieldName = 'JOB_CODE' Required = True Size = 5 end  <>object SmallintField2: TSmallintField  DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE' Required = True end  <>object StringField6: TStringField  DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end  <>object FloatField1: TFloatField
<pre>&lt;&gt;object StringField4: TStringField DisplayLabel = 'Dept No' FieldName = 'DEPT_NO' Required = True Size = 3 end &lt;&gt;object StringField5: TStringField DisplayLabel = 'Job Code' FieldName = 'JOB_CODE' Required = True Size = 5 end &lt;&gt;object SmallintField2: TSmallintField DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE' Required = True end &lt;&gt;object StringField6: TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end &lt;&gt;object FloatField1: TFloatField</pre>
TStringField  DisplayLabel = 'Dept No' FieldName = 'DEPT_NO' Required = True Size = 3 end <>object StringField5: TStringField DisplayLabel = 'Job Code' FieldName = 'JOB_CODE' Required = True Size = 5 end <>object SmallintField2: TSmallintField DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE' Required = True end <>object StringField6: TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end <>object FloatField1: TFloatField
DisplayLabel = 'Dept No' FieldName = 'DEPT_NO' Required = True Size = 3 end <>object StringField5: TStringField DisplayLabel = 'Job Code' FieldName = 'JOB_CODE' Required = True Size = 5 end <>object SmallintField2: TSmallintField DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE' Required = True end <>object StringField6: TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end <>object FloatField1: TFloatField
FieldName = 'DEPT_NO' Required = True Size = 3 end <>object StringField5: TStringField DisplayLabel = 'Job Code' FieldName = 'JOB_CODE' Required = True Size = 5 end <>object SmallintField2: TSmallintField DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE' Required = True end <>object StringField6: TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end <>object FloatField1: TFloatField
Required = True Size = 3 end <>object StringField5: TStringField DisplayLabel = 'Job Code' FieldName = 'JOB_CODE' Required = True Size = 5 end <>object SmallintField2: TSmallintField DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE' Required = True end <>object StringField6: TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end <>object FloatField1: TFloatField
size = 3 end  <>object StringField5: TStringField  DisplayLabel = 'Job Code' FieldName = 'JOB_CODE'  Required = True Size = 5 end  <>object SmallintField2: TSmallintField  DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE'  Required = True end  <>object StringField6: TStringField  DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end  <>object FloatField1: TFloatField
end <>object StringField5: TStringField DisplayLabel = 'Job Code' FieldName = 'JOB_CODE' Required = True Size = 5 end <>object SmallintField2: TSmallintField DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE' Required = True end <>object StringField6: TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end <>object FloatField1: TFloatField
<pre>&lt;&gt;object StringField5: TStringField DisplayLabel = 'Job Code' FieldName = 'JOB_CODE' Required = True Size = 5 end &lt;&gt;object SmallintField2: TSmallintField DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE' Required = True end &lt;&gt;object StringField6: TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end &lt;&gt;object FloatField1: TFloatField</pre>
TStringField  DisplayLabel = 'Job Code' FieldName = 'JOB_CODE' Required = True Size = 5 end <>object SmallintField2: TSmallintField DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE' Required = True end <>object StringField6: TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end <>object FloatField1: TFloatField
DisplayLabel = 'Job Code' FieldName = 'JOB_CODE' Required = True Size = 5 end <>object SmallintField2: TSmallintField DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE' Required = True end <>object StringField6: TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end <>object FloatField1: TFloatField
FieldName = 'JOB_CODE'  Required = True  Size = 5 end  <>object SmallintField2: TSmallintField  DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE'  Required = True end  <>object StringField6: TStringField  DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY'  Required = True Size = 15 end  <>object FloatField1: TFloatField
FieldName = 'JOB_CODE'  Required = True  Size = 5 end  <>object SmallintField2: TSmallintField  DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE'  Required = True end  <>object StringField6: TStringField  DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY'  Required = True Size = 15 end  <>object FloatField1: TFloatField
Size = 5 end  <>object SmallintField2: TSmallintField  DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE' Required = True end <>object StringField6: TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end <>object FloatField1: TFloatField
end <>object SmallintField2: TSmallintField  DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE' Required = True end <>object StringField6: TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end <>object FloatField1: TFloatField
<pre>&lt;&gt;object SmallintField2: TSmallintField DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE' Required = True end &lt;&gt;object StringField6: TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end &lt;&gt;object FloatField1: TFloatField</pre>
TSmallintField DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE' Required = True end <>object StringField6: TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end <>object FloatField1: TFloatField
TSmallintField DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE' Required = True end <>object StringField6: TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end <>object FloatField1: TFloatField
DisplayLabel = 'Job Grade' FieldName = 'JOB_GRADE' Required = True end <>object StringField6: TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end <>object FloatField1: TFloatField
FieldName = 'JOB_GRADE'  Required = True  end  <>object StringField6: TStringField  DisplayLabel = 'Job Country'  FieldName = 'JOB_COUNTRY'  Required = True  Size = 15  end  <>object FloatField1: TFloatField
Required = True end <>object StringField6: TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end <>object FloatField1: TFloatField
end <>object StringField6: TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end <>object FloatField1: TFloatField
<>object StringField6: TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end <>object FloatField1: TFloatField
TStringField DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end <>object FloatField1: TFloatField
DisplayLabel = 'Job Country' FieldName = 'JOB_COUNTRY' Required = True Size = 15 end <>object FloatField1: TFloatField
FieldName = 'JOB_COUNTRY'  Required = True  Size = 15  end  <>object FloatField1:  TFloatField
Required = True Size = 15 end <>object FloatField1: TFloatField
Size = 15 end <>object FloatField1: TFloatField
end <>object FloatField1: TFloatField
<>object FloatField1: TFloatField
TFloatField
Display I abol - !Calamy!
DisplayLabel = 'Salary'
FieldName = 'SALARY'
Required = True
end
<>object StringField7:
TStringField No. 10 No.
DisplayLabel = 'Name'
FieldName = 'FULL_NAME'
Size = 37
end
end
End

# (Test program for ADO Components)

مُوقعه: دلفي ٧ ( النسخة الأصلية )

الملف النصى والتنفيذي

#### **Procedures159**

unit AdoMain;
{ Test program for ADO
Components }
interface
uses
Variants, Windows, Sysutils,
Forms, IniFiles, ImgList, Controls,
Classes,
ActnList, Menus, Dialogs,
ComCtrls, ComObj, ToolWin, DB,
ADOInt,
Grids, DBGrids, Provider,
ADODB, DBClient, DBCtrls,
ExtCtrls, StdCtrls, Buttons, SqlEdit;
Stuctris, Buttons, SqiEuit;
type
type TADODBTest = class(TForm)
Connection: TADOConnection;
MasterTable: TADOTable;
DetailTable: TADOTable;
MasterQuery: TADOQuery;
DetailQuery: TADOQuery;
MasterProc: TADOStoredProc;
ADODataSet: TADODataSet;
Provider: TDataSetProvider;
MasterClientData:
TClientDataSet;
MasterDataSource: TDataSource;
DetailDataSource: TDataSource;
DetailGrid: TDBGrid;
MasterGrid: TDBGrid;
DBMemo1: TDBMemo;
DBImage1: TDBImage;
{ Actions }
ActionList1: TActionList;
SaveToFile: TAction;

البرنامج الشامل لتعليمات ADO
AdoTesto
اعداد:
علاء الدين محمد اللباد
.92200771
اسم البرنامج:

F: IN
FindNext: TButton;
Filtered: TCheckBox;
IndexFields: TEdit;
PrevQuery1: TSpeedButton;
Events: TListBox;
DescFields: TEdit;
CaseInsFields: TEdit;
MasterTableName: TComboBox;
DetailTableName: TComboBox;
MasterSQL: TMemo;
DetailSQL: TMemo;
GridSplitter: TSplitter;
ClearEventsButton: TToolButton;
LocateEdit: TEdit;
LocateField: TComboBox;
locPartialKey: TCheckBox;
LocateNull: TCheckBox;
UseClientCursorItem:
TMenuItem;
UseadCmdTableDirect1:
TMenuItem;
CursorTypeItem: TMenuItem;
CurTypeKeyset: TMenuItem;
Dynamic1: TMenuItem;
CurTypeUnspecified:
TMenuItem;
CurTypeForwardOnly: TMenuItem;
,
CurTypeStatic: TMenuItem;
DBEditScroller: TScrollBox;
LockTypeItem: TMenuItem;
LckTypeUnspecified:
TMenuItem;
LckTypeReadOnly: TMenuItem;
LckTypePessimistic: TMenuItem;
LckTypeOptimistic: TMenuItem;
LckTypeBatchOptimistic:
TMenuItem;
ReadOnlyLabel: TLabel;
ConnectionString: TComboBox;
EditConnStr: TSpeedButton;
Label1: TLabel;
Label2: TLabel;
ProcedureNames: TGroupBox;
TableNames: TGroupBox;
QueryStrings: TGroupBox;
MasterProcName: TComboBox;
Splitter1: TSplitter;
<b>Ֆրният, таришт,</b>

OpenQuery: TAction;
OpenTable: TAction;
BatchUpdate: TAction;
ExitApplication: TAction;
CloseActiveDataSet: TAction;
LoadFromFile: TAction;
CancelBatch: TAction;
ExecuteCommand: TAction;
StreamFormOut: TAction;
StreamFormIn: TAction;
ClearField: TAction;
ViewEvents: TAction;
PrevQuery: TAction;
NextQuery: TAction;
RefreshData: TAction;
ClearEventLog: TAction;
DisplayDetails: TAction;
HelpAbout: TAction;
UseClientCursor: TAction;
UseTableDirect: TAction;
,
UseShapeProvider: TAction; AsyncConnect: TAction;
·
AsyncExecute: TAction;
AsyncFetch: TAction;
OpenProcedure: TAction; MainMenu1: TMainMenu;
FileReopen: TMenuItem;
FileMenu: TMenuItem;
PopupMenu1: TPopupMenu;
ToolBar1: TToolBar;
ImageList1: TImageList;
OpenDialog: TOpenDialog;
SaveDialog: TSaveDialog;
StatusBar: TStatusBar;
AreaSelector: TPageControl;
DataPanel: TPanel;
FilterPage: TTabSheet;
LocatePage: TTabSheet;
IndexPage: TTabSheet;
FieldsPage: TTabSheet;
SourcePage: TTabSheet;
IndexList: TListBox;
NavigatorPanel: TPanel;
BlobCtrlPanel: TPanel;
GridPanel: TPanel;
DBNavigator1: TDBNavigator;
Filter: TEdit;
FindFirst: TButton;

TDv44ov	Data: IMastar Carrea
TButton;	DetailMasterSource: TDataSource;
ParameterType: TComboBox;	,
ToolButton3: TToolButton;	DetailQuerySource: TDataSource;
MidasApplyUpdatesButton:	CloseConnection: TAction;
TToolButton;	Disconnect1: TMenuItem;
MidasApplyUpdates: TAction;	BatchUpdates1: TMenuItem;
ADOButton: TRadioButton;	CancelBatch1: TMenuItem;
MidasButton: TRadioButton;	BatchUpdateButton:
MidasCancelUpdates: TAction;	TToolButton;
MidasCancelButton:	CancelBatchButton:
TToolButton;	TToolButton;
ApplyUpdatesMidas1:	AsyncConnect1: TMenuItem;
TMenuItem;	AsyncExecute1: TMenuItem;
CancelUpdatesMidas1:	AsyncFetch1: TMenuItem;
TMenuItem;	ADOCommand:
N6: TMenuItem;	TADOCommand;
SQLParams: TRadioButton;	ProgressBar: TProgressBar;
ProcParams: TRadioButton;	MaxRecords: TAction;
TestButton: TButton;	MaxRecords1: TMenuItem;
FieldSchemaGrid: TDBGrid;	DetailProcName: TComboBox;
FieldSchemaSource:	DetailProc: TADOStoredProc;
TDataSource;	ToolButton1: TToolButton;
FieldSchema: TADODataSet;	OpenProcedure1: TMenuItem;
FieldSchemaCOLUMN_NAME:	DetailProcSource: TDataSource;
TWideStringField;	EditCommandText:
FieldSchemaDATA_TYPE:	TSpeedButton;
TWordField;	ParamPage: TTabSheet;
	ParameterList: TListBox;
FieldSchemaNUMERIC_PRECISIO	ParameterName: TEdit;
N: TWordField;	ParameterValue: TEdit;
	ParameterSize: TEdit;
FieldSchemaCHARACTER_MAXI	ParameterNameLabel: TLabel;
MUM_LENGTH: TIntegerField;	ParameterScale: TEdit;
FieldSchemaNUMERIC_SCALE:	ParameterPrecision: TEdit;
TSmallintField;	PTypeLabel: TLabel;
EnableBCD: TAction;	PValueLabel: TLabel;
EnableBCD1: TMenuItem;	PSizeLabel: TLabel;
DisconnectDataSet: TAction;	PScaleLabel: TLabel;
DisconnectDataSet1: TMenuItem;	PPrecisionLabel: TLabel;
DetailClientData: TClientDataSet;	ParameterDirectionGroup:
FilterGroupBox: TRadioGroup;	TRadioGroup;
BlobAsImage: TAction;	ParamAttributes: TGroupBox;
BlobfieldasImage1: TMenuItem;	PANullableCheckBox:
LoadBlobFromFile: TAction;	TCheckBox;
LoadBlobfromfile1: TMenuItem;	PASignedCheckBox: TCheckBox;
IndexOptions: TGroupBox;	PALongCheckBox: TCheckBox;
idxCaseInsensitive: TCheckBox;	AddParameterButton: TButton;
idxDescending: TCheckBox;	RefreshParametersButton:

	'I D ' TCI I D
StreamFormInClick(Sender:	idxPrimary: TCheckBox;
TObject);	idxUnique: TCheckBox;
<> procedure	
LoadFromFileExecute(Sender:	<> procedure
TObject);	FilterKeyPress(Sender: TObject;
<> procedure	var Key: Char);
SaveToFileExecute(Sender:	<> procedure
TObject);	FormCreate(Sender: TObject);
<> procedure	<> procedure
EditActionsUpdate(Sender:	FormDestroy(Sender: TObject);
TObject);	<> procedure
<> procedure	<u> </u>
=	MasterSQLKeyPress(Sender:
FieldsPageShow(Sender: TObject);	TObject; var Key: Char);
<> procedure	<> procedure
OpenQueryExecute(Sender:	IndexListClick(Sender: TObject);
TObject);	<> procedure
<> procedure	GridTitleClick(Column: TColumn);
<b>ExecSQLExecute(Sender: TObject)</b> ;	<> procedure
<> procedure	LocateButtonClick(Sender:
OpenTableExecute(Sender:	TObject);
TObject);	<> procedure
<> procedure	FindFirstClick(Sender: TObject);
BatchUpdateExecute(Sender:	<> procedure
TObject);	FilterExit(Sender: TObject);
<> procedure	<> procedure
MasterTableNameDropDown(Sende	<u> </u>
r: TObject);	DataSourceDataChange(Sender:
	TObject; Field: TField);
<> procedure	<> procedure
ConnectionStringClick(Sender:	DataSetAfterOpen(DataSet:
TObject);	TDataSet);
<> procedure	<> procedure
ConnectionStringKeyPress(Sender:	LocateFieldDropDown(Sender:
TObject; var Key: Char);	TObject);
<> procedure	<> procedure
FilteredClick(Sender: TObject);	FindNextClick(Sender: TObject);
<> procedure	<> procedure
FilterPageShow(Sender: TObject);	MasterTableNameClick(Sender:
<> procedure	TObject);
IndexPageShow(Sender: TObject);	<> procedure
<> procedure	PopupMenu1Popup(Sender:
ExitApplicationExecute(Sender:	Tobject);
TObject);	• //
<> procedure	<> procedure FieldSelect(Sendow TObicot)
<u> </u>	FieldSelect(Sender: TObject);
CloseActiveDataSetExecute(Sender:	<> procedure
TObject);	GridColEnter(Sender: TObject);
<> procedure	<> procedure
FileActionsUpdate(Sender:	StreamFormOutClick(Sender:
TObject);	TObject);
<> procedure	<> procedure
	1

DataSetNewRecord(DataSet:	MasterTableNameKeyPress(Sender:
TDataSet);	TObject; var Key: Char);
<> procedure	<> procedure
DataSetAfterPost(DataSet:	DetailTableNameClick(Sender:
TDataSet);	TObject);
<> procedure	<> procedure
DataSetAfterInsert(DataSet:	MasterTableAfterOpen(DataSet:
TDataSet);	TDataSet);
<> procedure	<> procedure
DataSetAfterEdit(DataSet:	MasterTableBeforeClose(DataSet:
TDataSet);	TDataSet);
<> procedure	<> procedure
DataSetAfterDelete(DataSet:	GridSetFocus(Sender: TObject);
TDataSet);	<> procedure
<> procedure	LocatePageShow(Sender: TObject);
DataSetAfterCancel(DataSet:	<> procedure
TDataSet);	LocateNullClick(Sender: TObject);
<> procedure	<> procedure
MasterQueryAfterOpen(DataSet:	DataSetAfterScroll(DataSet:
TDataSet);	TDataSet);
<> procedure	<> procedure
MasterQueryBeforeClose(DataSet:	
TDataSet);	DataSetBeforeCancel(DataSet:
77	TDataSet);
<> procedure	<pre>&lt;&gt; procedure Dut Supplies Clarification </pre>
CancelBatchExecute(Sender:	DataSetBeforeClose(DataSet:
TObject);	TDataSet);
<> procedure	<> procedure
ClearFieldExecute(Sender:	DataSetBeforeDelete(DataSet:
TObject);	TDataSet);
<> procedure	<> procedure
ViewEventsExecute(Sender:	DataSetBeforeEdit(DataSet:
TObject);	TDataSet);
<> procedure	<> procedure
DisplayDetailsExecute(Sender:	DataSetBeforeInsert(DataSet:
TObject);	TDataSet);
<> procedure	<> procedure
DataSourceStateChange(Sender:	DataSetBeforePost(DataSet:
TObject);	TDataSet);
<> procedure	<> procedure
DataSourceUpdateData(Sender:	DataSetBeforeScroll(DataSet:
TObject);	TDataSet);
<> procedure	<> procedure
RefreshDataExecute(Sender:	DataSetCalcFields(DataSet:
TObject);	TDataSet);
<> procedure	<> procedure
ClearEventLogExecute(Sender:	DataSetError(DataSet: TDataSet; E:
TObject);	EDatabaseError;
<> procedure	var Action: TDataAction);
ClearEventLogUpdate(Sender:	<> procedure
Cival Excite Schauchellaci.	procedure

<> procedure	TObject);
DetailQueryBeforeOpen(DataSet:	<> procedure
TDataSet);	HelpAboutExecute(Sender:
<> procedure	TObject);
MasterProcBeforeOpen(DataSet:	<> procedure
TDataSet);	DataSetAfterClose(DataSet:
<> procedure	TDataSet);
UseShapeProviderExecute(Sender:	<> procedure
TObject);	FileMenuClick(Sender: TObject);
<> procedure	
OnFilterRecord(DataSet: TDataSet;	<pre>&lt;&gt; procedure ClassdFileClick(Sandam TObiast);</pre>
	ClosedFileClick(Sender: TObject);
var Accept: Boolean);	<> procedure
<> procedure	FormCloseQuery(Sender: TObject;
BinaryGetText(Sender: TField; var	var CanClose: Boolean);
Text: string;	<> procedure
DisplayText: Boolean);	PrevQueryExecute(Sender:
<> procedure	TObject);
BinarySetText(Sender: TField; const	<> procedure
Text: string);	PrevQueryUpdate(Sender:
<> procedure	TObject);
ConnectionBeginTransComplete(Co	<> procedure
nnection: TADOConnection;	NextQueryExecute(Sender:
TransactionLevel: Integer; const	TObject);
Error: Error;	<> procedure
var EventStatus: TEventStatus);	MasterTableAfterScroll(DataSet:
<> procedure	TDataSet);
ConnectionCommitTransComplete(	<> procedure
Connection: TADOConnection;	MasterTableBeforeScroll(DataSet:
const Error: Error; var	TDataSet);
EventStatus: TEventStatus);	<> procedure
<> procedure	RadioItemClick(Sender: TObject);
ConnectionConnectComplete(Conne	<> procedure
ction: TADOConnection;	DataSetBeforeOpen(DataSet:
const Error: Error; var	TDataSet);
EventStatus: TEventStatus);	<> procedure
·	BooleanActionExecute(Sender:
<pre>&lt;&gt; procedure ConnectionDisconnect(Connection:</pre>	`
ConnectionDisconnect(Connection:	TObject);
TADOConnection;	<> procedure
var EventStatus: TEventStatus);	EditConnStrClick(Sender:
<> procedure	TObject);
ConnectionExecuteComplete(Conne	<> procedure
ction: TADOConnection;	MasterTableBeforeOpen(DataSet:
RecordsAffected: Integer; const	TDataSet);
Error: Error;	<> procedure
var EventStatus: TEventStatus;	DetailTableBeforeOpen(DataSet:
const Command: _Command;	TDataSet);
const Recordset: _Recordset);	<> procedure
<> procedure	MasterQueryBeforeOpen(DataSet:
ConnectionInfoMessage(Connection	TDataSet);
	<u> </u>

TObject);	: TADOConnection;
<> procedure	const Error: Error; var
ProcNameDropDown(Sender:	EventStatus: TEventStatus);
TObject);	<> procedure
<> procedure	ConnectionRollbackTransComplete(
MasterProcNameKeyPress(Sender:	<b>Connection: TADOConnection;</b>
TObject; var Key: Char);	const Error: Error; var
<> procedure	EventStatus: TEventStatus);
MasterProcNameClick(Sender:	<> procedure
TObject);	ConnectionWillConnect(Connection
<> procedure	: TADOConnection;
DetailProcNameClick(Sender:	var ConnectionString, UserID,
TObject);	Password: WideString;
<> procedure	var ConnectOptions:
OpenProcedureExecute(Sender:	TConnectOption; var EventStatus:
TObject);	TEventStatus);
<> procedure	,,
DetailProcBeforeOpen(DataSet:	<pre>&lt;&gt; procedure ConnectionWillEvecute(Connection)</pre>
TDataSet);	ConnectionWillExecute(Connection:
,,	TADOConnection;
<> procedure	var CommandText: WideString;
MasterProcAfterOpen(DataSet:	var CursorType: TCursorType;
TDataSet);	var LockType: TADOLockType;
<> procedure	var CommandType:
EditCommandTextClick(Sender:	TCommandType;
TObject);	var ExecuteOptions:
<> procedure	TExecuteOptions; var EventStatus:
ParamPageShow(Sender: TObject);	TEventStatus;
<> procedure	const Command: _Command;
RefreshParametersButtonClick(Sen	const Recordset: _Recordset);
der: TObject);	<> procedure
<> procedure	CloseConnectionExecute(Sender:
ParameterListClick(Sender:	TObject);
TObject);	<> procedure
<> procedure	DataSetFetchComplete(DataSet:
AddParameterButtonClick(Sender:	TCustomADODataSet;
TObject);	const Error: Error; var
<> procedure	EventStatus: TEventStatus);
ParameterDataChange(Sender:	<> procedure
TObject);	ExceptionHandler(Sender: TObject;
<> procedure	E: Exception);
MasterSQLChange(Sender:	<> procedure
TObject);	ConnectionLogin(Sender: TObject;
<> procedure	Username, Password: String);
MidasApplyUpdatesExecute(Sender	<> procedure
: TObject);	MasterGridColumnMoved(Sender:
<> procedure	TObject; FromIndex,
ADOButtonClick(Sender: TObject);	ToIndex: Integer);
<> procedure	<> procedure
MidasButtonClick(Sender:	MaxRecordsExecute(Sender:

TC 4 ADOD 4 C 4	TOL: ()
TCustomADODataSet;	TObject);
FActiveDataSource:	<> procedure
TDataSource;	MasterClientDataReconcileError(D
FStatusMsg: string;	ataSet: TCustomClientDataSet;
FClosedTables: TStringList;	E: EReconcileError;
FMasterQueries: TStringList;	UpdateKind: TUpdateKind;
FDetailQueries: TStringList;	var Action: TReconcileAction);
FQueryIndex: Integer;	<> procedure
FModifiedParameter: Integer;	MidasCancelUpdatesExecute(Sende
FMovingColumn: Boolean;	r: TObject);
FParamSource: TParameters;	<> procedure
FLastDataFile: String;	ParameterSourceClick(Sender:
FLastFormFile: String;	TObject);
The series of the series,	<> procedure
function CotConfigEiler TiniEiler	FieldSchemaDATA TYPEGetText(
function GetConfigFile: TIniFile;	Sender: TField; var Text: String;
I	DisplayText: Boolean);
RefreshIndexNames;	<> procedure
<pre>&lt;&gt; procedure Set Active Data Set (Value: TData Set);</pre>	DisconnectDataSetExecute(Sender:
SetActiveDataSet(Value: TDataSet);	TObject);
<> procedure	<> procedure
SetEventsVisible(Visible: Boolean);	FieldValidate(Sender: TField);
<> procedure SetQueryText;	<> procedure
<> procedure	TestButtonClick(Sender: TObject);
SetStatusMsg(const Msg: string);	
<> procedure	<pre>&lt;&gt; procedure DataSatEatah Programmas(DataSate)</pre>
ShowHeapStatus(Sender: TObject;	DataSetFetchProgress(DataSet:
var Done: Boolean);	TCustomADODataSet; Progress,
<> procedure	MaxProgress: Integer; var
UpdateReOpenMenu;	EventStatus: TEventStatus);
<> procedure	<> procedure
OnHint(Sender: TObject);	FilterGroupBoxClick(Sender:
<> procedure	TObject);
ClearProgressBar;	<> procedure
<> procedure	BlobAsImageUpdate(Sender:
ShowProgressBar(const Msg:	TObject);
string);	<> procedure
<> procedure	BlobAsImageExecute(Sender:
ProcessQuery(SelectQuery:	TObject);
Boolean);	<> procedure
<> procedure	LoadBlobFromFileExecute(Sender:
WriteParameterData;	TObject);
<> procedure	<> procedure
UpdateParameterList;	ToolBar1Click(Sender: TObject);
<> procedure	private
ShowIndexParams;	FConfig: TIniFile;
<> procedure	FMaxErrors: Integer;
SetRecordSetEvents(Hook: Boolean;	FPacketRecs: Integer;
DataSet: TCustomADODataSet);	FActiveDataSet: TDataSet;
function GetActiveDataSet:	FADOSource:
tunction GetActiveDataSet:	TIE OSOUICE!

begin	TDataSet;
F := CreateMessageDialog(",	<i>'</i>
8 8	public <> procedure
mtInformation, [mbCancel]);	F
F.Height := Screen.Height div 2;	BindControls(DataSet: TDataSet);
F.Width := Screen.Width div 2;	<> procedure
Button := F.Components[2] as	CheckConnection(CloseFirst:
TButton;	Boolean = False);
Button.Top := F.ClientHeight -	<> procedure
Button.Height - 5;	OpenDataSet(Source:
Button.Left := (F.ClientWidth -	TCustomADODataSet);
Button.Width) div 2;	<> procedure
F.Caption := 'Properties';	StreamSettings(Write: Boolean);
with TMemo.Create(F) do	<> procedure LogEvent(const
begin	EventStr: string; Component:
SetBounds(5, 5, F.ClientWidth-10,	TComponent = nil);
F.ClientHeight - 40);	<> procedure
Parent := F;	RefreshParameters(Parameters:
for I := 0 to Props.Count - 1 do	TParameters);
with Props[I] do	property StatusMsg: string read
Lines.Add(Format('%-30s: %s',	FStatusMsg write SetStatusMsg;
[Name, VarToStr(Value)]));	property ActiveDataSet: TDataSet
end;	read GetActiveDataSet write
F.ShowModal;	SetActiveDataSet;
end;	property ActiveDataSource:
	TDataSource read
{\$R *.dfm}	FActiveDataSource write
	FActiveDataSource;
procedure	property ADOSource:
TADODBTest.FormCreate(Sender:	TCustomADODataSet read
TObject);	FADOSource write FADOSource;
9 //	property ConfigFile: TIniFile read
procedure SetupControls;	GetConfigFile;
var	end;
I: Integer;	
begin	var
for I := 0 to	ADODBTest: TADODBTest;
StatusBar.Panels.Count - 1 do	
StatusBar.Panels[I].Text := ";	implementation
ProgressBar.Parent := StatusBar;	
ProgressBar.SetBounds(0, 2,	uses
StatusBar.Panels[0].Width,	OLEDB, DBLogDlg, AdoConEd,
StatusBar. Failets[0]. Wittil, StatusBar. Height - 2);	RecError;
{ Set these dynamically since the	·
` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` `	procedure ShowProperties(Props:
form may have been scaled!	
form may have been scaled }	1 - 1
DataPanel.Constraints.MinWidth	Properties);
,	Properties); var
DataPanel.Constraints.MinWidth := DataPanel.Width;	Properties); var I: Integer;
DataPanel.Constraints.MinWidth	Properties); var

Application.Terminate;	:= AreaSelector.Width;
end;	Constraints.MinHeight := Height -
ciu,	(DataPanel.Height -
procedure	DataPanel.Constraints.MinHeight);
TADODBTest.HelpAboutExecute(Se	Datai anci. Consti antis. Minificignity,
nder: TObject);	SetEventsVisible(ViewEvents.Check
begin	ed);
begin	end;
ShowMassaga(Cantion±#13#10±1Ca	ciiu,
ShowMessage(Caption+#13#10+'Copyright (c) 1999-2002 Borland	begin
Corporation');	FMaxErrors := -1;
end;	FPacketRecs := -1;
chu,	FModifiedParameter := -1;
procedure	ActiveDataSource :=
TADODBTest.OnHint(Sender:	MasterDataSource;
TObject);	Waster DataSource,
begin	SetCurrentDirectory(PChar(Extract
if	FilePath(ParamStr(0))));
FindVCLWindow(Mouse.CursorPos	Application.OnIdle :=
) <> ConnectionString then	ShowHeapStatus;
ConnectionString.Hint :=	Application.OnHint := OnHint;
ConnectionString.Text;	FClosedTables :=
StatusMsg := Application.Hint;	TStringList.Create;
end;	FMasterQueries :=
ciiu,	TStringList.Create;
procedure	FDetailQueries :=
TADODBTest.ExceptionHandler(Se	TStringList.Create;
nder: TObject; E: Exception);	StreamSettings(False);
begin	SetupControls;
ClearProgressBar;	ParameterSourceClick(Self);
Cicari rogicssbar,	end;
SysUtils.ShowException(ExceptObje	enu,
ct, ExceptAddr);	procedure
end;	procedure TADODBTest.FormDestroy(Sender:
· · · · · · · · · · · · · · · · · · ·	TObject);
{ View Options }	begin
( . zerr operons )	if Assigned(FConfig) then
procedure	StreamSettings(True);
TADODBTest.SetEventsVisible(Visi	FConfig.Free;
ble: Boolean);	FDetailQueries.Free;
var	FMasterQueries.Free;
EventsWidth: Integer;	FClosedTables.Free;
begin	end;
Constraints.MinWidth := 0;	ciiu,
if Events.Visible <> Visible then	procedure
begin	procedure TADODPTost Evit Application Even
DataPanel.Anchors :=	TADODBTest.ExitApplicationExecu
	te(Sender: TObject);
DataPanel.Anchors - [akRight];	begin
AreaSelector.Anchors :=	

hogin	Avos Coloston Anchons [als Dight]
begin	AreaSelector.Anchors - [akRight];
if FConfig = nil then	try
FConfig :=	EventsWidth := Events.Width +
TIniFile.Create(ChangeFileExt(Para	5;
mStr(0), '.INI'));	Events. Visible := Visible;
Result := FConfig;	if not Visible then
end;	EventsWidth := -EventsWidth;
	ClientWidth := ClientWidth +
procedure	EventsWidth;
TADODBTest.StreamSettings(Write	finally
: Boolean);	DataPanel.Anchors :=
	DataPanel.Anchors + [akRight];
procedure WriteStr(const	AreaSelector.Anchors :=
OptName: string; Value: Variant);	AreaSelector.Anchors + [akRight];
begin	end;
FConfig.WriteString('Settings',	end;
OptName, Value);	if Visible then
end;	Constraints.MinWidth :=
	DataPanel.Constraints.MinWidth +
procedure WriteBool(const	Events.Width + 22 else
OptName: string; Value: Boolean);	Constraints.MinWidth :=
begin	DataPanel.Constraints.MinWidth +
FConfig.WriteBool('Settings',	18;
OptName, Value);	end;
end;	
	procedure
procedure WriteStrings(const	TADODBTest.ViewEventsExecute(S
SectName: string; Values: TStrings);	ender: TObject);
var	begin
I: Integer;	ViewEvents.Checked := not
begin	ViewEvents.Checked;
FConfig.EraseSection(SectName);	view Events entered,
for I := 0 to Values.Count - 1 do	SetEventsVisible(ViewEvents.Check
FConfig.WriteString(SectName,	ed);
IntToStr(I), Values[I]);	end;
	ciu,
end;	procedure
for day Dec 184 / 4 O AN	procedure TADODBTest.DisplayDetailsExecut
function ReadStr(const OptName:	= 7
string): Variant;	e(Sender: TObject);
begin	begin  DisplayDatails Chasked - not
Result :=	DisplayDetails.Checked := not
FConfig.ReadString('Settings',	DisplayDetails.Checked;
OptName, ");	end;
end;	(0,4)
	{ Settings }
function ReadBool(const OptName:	
string): Boolean;	function
begin	TADODBTest.GetConfigFile:
	TIniFile;

with TEdit(Components[I]) do
WriteStr(Name, Text)
else if Components[I] is
TComboBox then
with
TDBComboBox(Components[I]) do
WriteStr(Name, Text)
else if Components[I] is
TCheckBox then
with
TCheckBox(Components[I]) do
WriteBool(Name, Checked)
else if Components[I] is
TRadioButton then
with
TRadioButton(Components[I]) do
WriteBool(Name, Checked)
else if Components[I] is TAction
then
with TAction(Components[I])
do
WriteBool(Name, Checked)
else if Components[I] is
TPageControl then
with
TPageControl(Components[I]) do
WriteStr(Name,
ActivePage.Caption)
else if Components[I] is
TMenuItem then
with
TMenuItem(Components[I]) do
for J := 0 to Count-1 do
if Items[J].Checked then
begin
WriteStr(Name, J);
System.Break;
end;
end
else
begin
for I := Low(Components) to
High(Components) do
if Components[I] is
TCustomEdit then
with TEdit(Components[I]) do
Text := ReadStr(Name)
else if Components[I] is
3.50 1. 0011-011-011-15

Result :=
FConfig.ReadBool('Settings',
OptName, False);
end;
procedure ReadStrings(const
SectName: string; Values: TStrings);
var
I: Integer;
S: string;
begin
for I := 0 to 99 do
begin
S :=
FConfig.ReadString(SectName,
IntToStr(I), ");
if S = " then Break;
Values.Add(S);
end;
end;
function FindPage(const
PageName: string): TTabSheet;
var
I: Integer;
begin
for I := AreaSelector.PageCount -
1 downto 0 do
begin
Result := AreaSelector.Pages[I];
if Result.Caption = PageName
then Exit;
end;
Result := SourcePage;
end;
procedure
ProcessComponents(Components:
array of TComponent);
var
I,J: Integer;
begin
if Write then
begin
for I := Low(Components) to
High(Components) do
if Components[I] is
TCustomEdit then

[	
MasterProcName, DetailProcName,	TComboBox then
MasterSQL, DetailSQL,	with
ViewEvents, DisplayDetails,	TComboBox(Components[I]) do
UseClientCursor, UseTableDirect,	Text := ReadStr(Name)
UseShapeProvider,	else if Components[I] is
CursorTypeItem,	TCheckBox then
LockTypeItem, AsyncConnect,	with
AsyncExecute, AsyncFetch,	TCheckBox(Components[I]) do
MidasButton,	Checked := ReadBool(Name)
ProcParams, EnableBCD]);	else if Components[I] is
if Write then	TRadioButton then
begin	with
WriteStrings('ConnectionStrings',	TRadioButton(Components[I]) do
ConnectionString.Items);	Checked := ReadBool(Name)
WriteStrings('ClosedTables',	else if Components[I] is TAction
FClosedTables);	then
WriteStrings('MasterQueries',	with TAction(Components[I])
FMasterQueries);	do
WriteStrings('DetailQueries',	Checked := ReadBool(Name)
FDetailQueries);	else if Components[I] is
FConfig.UpdateFile;	TPageControl then
end else	with
begin	TPageControl(Components[I]) do
ReadStrings('ConnectionStrings',	ActivePage :=
ConnectionString.Items);	FindPage(ReadStr(Name))
ReadStrings('ClosedTables',	else if Components[I] is
FClosedTables);	TMenuItem then
ReadStrings('MasterQueries',	with
FMasterQueries);	TMenuItem(Components[I]) do
ReadStrings('DetailQueries',	(
FDetailQueries);	Items[ReadStr(Name)].Checked :=
end;	True;
end;	end;
	end;
procedure	
TADODBTest.RadioItemClick(Send	begin
er: TObject);	GetConfigFile;
begin	if not Write and
(Sender as TMenuItem).Checked	(ReadStr('AreaSelector') = '') then
:= True;	begin
end;	ConnectionString.Text := 'FILE
· · · · · · · · · · · · · · · · · · ·	NAME=' + DataLinkDir +
procedure	'\DBDEMOS.UDL';
TADODBTest.BooleanActionExecut	Exit;
e(Sender: TObject);	end;
begin	,
TAction(Sender).Checked := not	ProcessComponents([AreaSelector,
,	ConnectionString,
TAction(Sender).Checked;	MasterTableName,
end;	DetailTableName,

C( / D D 1/21/E /	
StatusBar.Panels[3].Text :=	
Msg+'';	procedure
while ProgressBar.Visible do	TADODBTest.UseShapeProviderEx
begin	ecute(Sender: TObject);
ProgressBar.StepIt;	begin
Application.ProcessMessages;	BooleanActionExecute(Sender);
Sleep(ProgressBar.Position);	Connection.Close;
end;	end;
end;	
	procedure
procedure	TADODBTest.MaxRecordsExecute(
TADODBTest.ClearProgressBar;	Sender: TObject);
begin	var
ProgressBar.Hide;	MaxRecs: string;
ProgressBar.Position := 0;	begin
StatusBar.Panels[3].Text := ";	MaxRecs :=
end;	IntToStr(MaxRecords.Tag);
	if InputQuery(Application.Title,
procedure	MaxRecords.Hint, MaxRecs) then
TADODBTest.DataSourceDataChan	MaxRecords.Tag :=
ge(Sender: TObject;	StrToInt(MaxRecs);
Field: TField);	end;
const	· · ·
StatusStrs: array[TUpdateStatus]	{ Status Information }
of string = ('Unmodified',	(Status Information)
'Modified', 'Inserted', 'Deleted');	procedure
	TADODBTest.ShowHeapStatus(Sen
begin	der: TObject; var Done: Boolean);
if (Sender = ActiveDataSource) and	begin
Assigned(ActiveDataSource.DataSet	Caption := Format('ADO DB
) and	Controls Test Application -
ActiveDataSource.DataSet.IsSequen	
ced then	(Blocks=%d Bytes=%d)',
begin	[AllocMemCount,
with ActiveDataSource.DataSet do	AllocMemSize]);
begin	end;
if IsEmpty then	1
begin	procedure
StatusBar.Panels[1].Text := ";	TADODBTest.SetStatusMsg(const
StatusBar.Panels[3].Text :=	Msg: string);
'(empty)';	begin
end else	StatusBar.Panels[0].Text := Msg;
begin	end;
StatusBar.Panels[1].Text :=	
StatusStrs[UpdateStatus];	procedure
if (State = dsBrowse) and (Field	TADODBTest.ShowProgressBar(co
= nil) then	nst Msg: string);
begin	begin
StatusBar.Panels[3].Text :=	ProgressBar.Show;
	· · · · · · · · · · · · · · · · · · ·

StatusMsg := Format('Orignal Value: %s', [VarToStr(V)]) else   StatusMsg := ";   end;   end;   end;   end;   if ActiveControl is TDBGrid then	TO LIKE I A	T (10/ 1 00/ 11 ID N
Value: %s',  VarToStr(V) ) else  StatusMsg := "; end; end;  begin  Field := (Sender as TDBGrid).SelectedField; if Assigned(Field) then begin  (Sender as TDBGrid).Hint := Field.ClassName; StatusBar.Panels[0].Text := Field.ClassName; StatusBar.Panels[2].Text := NullStr Field.IshNull ; TrackBlobs; if ActiveDataSet.UpdateStatus = usModified then  ShowOriginalValues; end; end;  { Connection Operations }  Connection Operations }  const  ConnectOptionValues: array Boolean] of TConnectOption = (coConnectUnspecified, coAsyneConnect); var  ConnStr: string; Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.Connected then Exit; Connection.ConnectonString := ConnStr: Connection.ConnectOptions :=  Connection.ConnectOptions :=  Connection.ConnectOptions :=	Field.Value) then	Format('%d of %d', [RecNo,
end; end; end; begin Field := (Sender as TDBGrid).SelectedField; if Assigned(Field) then begin (Sender as TDBGrid).Hint := Field.ClassName; StatusBar.Panels[0].Text := Field.ClassName; StatusBar.Panels[2].Text := NullStr[Field.IsNull]; TrackBlobs; if ActiveDotaSet.UpdateStatus = usModified then ShowOriginalValues; end; end; end; end;  { Connection Operations }  Field. Tield. DataSet. GridColEnter(Sender : TObject); const NullStr: array[Boolean] of string = (".'[NULL]'); var Field. Tield;  Fried. Tield;  procedure TADODBTest. CheckConnection(CloseFirst: Boolean); const ConnectOptionValues: array [Boolean] of TConnectOption = (coConnectUnspecified, coAsyncConnect); var Connstr: string; Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.Connected then Exit; Connection.ConnectionString := ConnStr: Connection.ConnectOptions :=  Connection.ConnectOptions :=  Connection.ConnectOptions :=		17/
end; end; begin Ficld := (Sender as TDBGrid).SelectedField; if Assigned(Field) then begin (Sender as TDBGrid).Hint := Field.ClassName; StatusBar.Panels[0].Text := Field.ClassName; StatusBar.Panels[2].Text := NullStr[Field.IsNull]; TrackBlobs; if ActiveDataSet.UpdateStatus = usModified then ShowOriginalValues; end; end; end;  {Connection Operations }  Field. TrickBlobs; if ActiveDataSet.UpdateStatus = usModified then ShowOriginalValues; end; end; frield.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Field.Fiel		
if ActiveControl is TDBGrid then  begin  Field := (Sender as TDBGrid).SelectedField; if Assigned(Field) then begin  (Sender as TDBGrid).Hint := Field.ClassName; StatusBar.Panels[0].Text := Field.ClassName; StatusBar.Panels[2].Text := NullStr[Field.IsNull]; TrackBlobs; if ActiveDataSet.UpdateStatus = usModified then  ShowOriginalValues; end; end; end;  end;  { Connection Operations }  Connection Operations }  ConnectOptionValues: array   Boolean   of TConnectOption = (coConnectUnspecified, coAsyncConnect); var  Connstr: string; Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.Close; MasterClientData.Close; Connection.Connectontonstring := ConnStr; Connection.ConnectOptions :=   if ActiveControl is TDBGrid then  GridColEnter(TDBGrid(ActiveCont rol)); end; end;  cond; end;   Cogewent('OnDataChange', Sender as TComponent); end;   Field.TackBlobs; end;   TADODBTest.GridColEnter(Sender as TComponent); end;   Field: Trield;   procedure TackBlobs; begin if Field.DataSet ⇔ MasterDataSource.DataSet then Exit;  if (Field is TMemoField) and (Field ⇔ DBImage1.Field) then DBImage1.DataField := Field.FieldName; end;   procedure ShowOriginalValues;  var  V: Variant; begin if Field.Dataset.CanModify then begin   V:= Field.OldValue;		,
begin Field := (Sender as TDBGrid).SelectedField; if Assigned(Field) then begin (Sender as TDBGrid).Hint := Field.ClassName; StatusBar.Panels[0].Text := Field.ClassName; StatusBar.Panels[2].Text := NullStr[Field.IsNull]; TrackBlobs; if ActiveDataSet.UpdateStatus = usModified then ShowOriginalValues; end; end;  { Connection Operations }  frocedure TADODBTest.CheckConnection(CloseFirst: Boolean); const ConnectOptionValues: array [Boolean] of TConnectOption = (coConnectUnspecified, coAsyncConnect); var ConnStr: string; Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.ConnectionString:= ConnStr: = ConnectionString.Text; Connection.ConnectOptions :=  then  GridColEnter(TDBGrid(ActiveCont roll); end; end;  LogEvent('OnDataChange', Sender as TComponent); end; TADODBTest.GridColEnter(Sender : TObject); const NullStr: array[Boolean] of string = (",','NULL1'); var Field.TField;  procedure TrackBlobs; begin if Field.DataSet ⇔ MasterDataSource.DataSet then Exit; if (Field is TMemoField) and (Field ⇔ DBMemol.DataField := Field.FieldName else if (Field is TGraphicField) and (Field ⇔ DBImage1.DataField := Field.FieldName; else if (Field is TGraphicField) and (Field ⇔ DBImage1.DataField := Field.FieldName; else if (Field is TGraphicField) and (Field ⇔ DBImage1.DataField := Field.FieldName; else if (Field is TGraphicField) and (Field ⇔ DBImage1.DataField := Field.FieldName; else if (Field is TGraphicField) and (Field ⇔ DBImage1.DataField := Field.FieldName; else if (Field is TGraphicField) and (Field ⇔ DBImage1.DataField := Field.FieldName; else if (Field is TGraphicField) and (Field ⇔ DBImage1.DataField := Field.FieldName; else if (Field is TGraphicField) and (Field ⇒ DBImage1.DataField := Field.FieldName; else if (Field is TGraphicField) and (Field ⇒ DBImage1.DataField := Field.FieldName; else if (Field is TGraphicField) and (Field ⇒ DBImage1.DataField := Field.FieldName; else if (Field is TGraphicField) and (Field ⇒ DBImage1.Field) then DBImage1.DataField := Field.FieldName; else if (Fi	end;	,
Degin	end;	if ActiveControl is TDBGrid
Field := (Sender as TDBGrid).SelectedField; if Assigned(Field) then begin     (Sender as TDBGrid).Hint := Field.ClassName;     StatusBar.Panels[0].Text := Field.ClassName;     StatusBar.Panels[2].Text := NullStr[Field.IsNull];     TrackBlobs; if ActiveDataSet.UpdateStatus = usModified then     ShowOriginalValues; end; end; end;  { Connection Operations }  procedure     TADODBTest.CheckConnection(CloseFirst: Boolean); const     ConnectOptionValues: array   Boolean   of TConnectOption = (coConnectUnspecified, coAsyncConnect); var     ConnectIonStr: string; Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.Connected then Exit; Connection.ConnectorString := Connstr: Connection.ConnectorString := Connstr: Connection.ConnectOptions := V := Field.OldValue;  GridColEnter(TDBGrid(ActiveCont roll); end; end; end;  LogEvent('OnDataChange', Sender as TComponent); end; end;  Index: TabODBTest.GridColEnter(Sender		then
TDBGrid).SelectedField; if Assigned(Field) then begin (Sender as TDBGrid).Hint := Field.ClassName; StatusBar.Panels[0].Text := Field.ClassName; StatusBar.Panels[2].Text := NullStr[Field.IsNull]; TrackBlobs; if ActiveDataSet.UpdateStatus = usModified then ShowOriginalValues; end; end;  { Connection Operations }  procedure TADODBTest.CheckConnection(CloseFirst: Boolean); const ConnectOptionValues: array [Boolean] of TConnectOption = (coConnectUnspecified, coAsyncConnect); var ConnStr: string; Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.ConnectionString.Text; Connection.ConnectionString:= ConnStr; Connection.ConnectOptions :=  Connection.ConnectOptions :=  rol)); end; end; cnd; LogEvent('OnDataChange', Sender as TComponent); end;  rol)); end; end;  cnd;  Field.Tield is TGrable is TGrable is Tield;  rol)); end; end;  cond;  TADODBTest.GridColEnter(Sender : TObject); const . TObject); const . NullStr: array[Boolean] of string = (".', NULL!"); var  Field. Tield;  procedure TrackBlobs; begin if Field.DataSet <> MasterDataSource.DataSet then Exit; if (Field is TMemoField) and (Field ◇ DBMemol.DataField := Field.FieldName; else if (Field is TGraphicField) and (Field ◇ DBImagel.Field) then  DBImagel.DataField := Field.FieldName; end;  if not CloseFirst and Connection.Connected then Exit; Connection.ConnectionString.Text; Connection.ConnectionString.Text; Connection.ConnectionString := ConnStr: Connection.ConnectOptions :=	begin	
if Assigned(Field) then begin  (Sender as TDBGrid).Hint := Field.ClassName; StatusBar.Panels[0].Text := Field.ClassName; StatusBar.Panels[2].Text := NullStr[Field.IsNull]; TrackBlobs; if ActiveDataSet.UpdateStatus = usModified then ShowOriginalValues; end; end; end;  { Connection Operations }  procedure TADODBTest.CheckConnection(CloseFirst: Boolean); const ConnectOptionValues: array [Boolean] of TConnectOption = (coConnectUnspecified, coAsyncConnect); var ConnStr: string; Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.ConnectonString.Text; Connection.ConnectonString.Text; Connection.ConnectOptions :=  Connstr; Connection.ConnectOptions :=  end; end; tageVent('OnDataChange', Sender as TComponent); end; cond;  tageVent('OnDataChange', Sender as TComponent); end; cond;  tageVent('OnDataChange', Sender as TComponent); end;  if Field.DataSet.Coletonetionent, if Field, DataSet.Call and (Field DataSet.Call and (Field DataSet.Call and (Field DataSet.Call and (Field DataSe	Field := (Sender as	GridColEnter(TDBGrid(ActiveCont
begin  (Sender as TDBGrid).Hint := Field.ClassName;  StatusBar.Panels[0].Text := Field.ClassName;  StatusBar.Panels[2].Text := NullStr[Field.IsNull];  TrackBlobs; if ActiveDataSct.UpdateStatus = usModified then ShowOriginalValues; end; end; end;  { Connection Operations }  procedure TADODBTest.CheckConnection(CloseFirst: Boolean) of TconnectOption = (coConnectUnspecified, coAsyncConnect); var  ConnStr: string; Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.ConnectionString.Text; Connection.ConnectionString.Text; Connection.ConnectOptions :=  end;  LogEvent('OnDataChange', Sender as TComponent); end;  TADODBTest.GridColEnter(Sender : TObject); const  NullStr: array[Boolean] of string = ("', [NULL]'); var  Field: TField;  procedure TrackBlobs; begin if Field.DataSet <> MasterDataSet then Exit; if (Field is TMemoField) and (Field <> DBMemo1.DataField := Field.FieldName else if (Field is TGraphicField) and (Field <> DBImage1.DataField := Field.FieldName; end;  procedure ShowOriginalValues; var  V: Variant; begin  V: Variant; begin  V: Field.OldValue;	TDBGrid).SelectedField;	rol));
cend;   cend;   component;	if Assigned(Field) then	end;
Csender as TDBGrid).Hint := Field.ClassName;   StatusBar.Panels[0].Text := Field.ClassName;   StatusBar.Panels[2].Text := NullStr[Field.IsNull];   TrackBlobs;   if ActiveDataSet.UpdateStatus = usModified then   ShowOriginalValues;   end;   end;     Endit TField;   StatusBar.CheckConnection(CloseFirst: Boolean)   of TConnectOption Values: array   Boolean   of TConnectOption = (coConnectUnspecified, coAsyncConnect);   var   Connstr: string;   Index: Integer;   begin   if not CloseFirst and   Connection.Connected then Exit;   Connection.ConnectionString.Text;   Connection.ConnectionString := Connection.ConnectOptions :=   Connection.ConnectOptions :=   V: Field.OldValue;		end;
Field.ClassName; StatusBar.Panels[0].Text := Field.ClassName; StatusBar.Panels[2].Text := NullStr[Field.IsNull]; TrackBlobs; if ActiveDataSet.UpdateStatus = usModified then ShowOriginalValues; end; end; end;  { Connection Operations }  Field: Trield;  { Connection Operations }  Field: Trield;  { Connection Operations }  Field: Trield;  Field: Trield;  Field.DataSet <> MasterDataSource.DataSet then Exit; if (Field is TMemoField) and (Field > DBMemo1.Field) then DBMemo1.DataField := Field.FieldName else if (Field is TGraphicField) and (Field > DBImage1.Field) then DBImage1.DataField := Field.FieldName; else if (Field is TGraphicField) and (Field > DBImage1.Field) then DBImage1.DataField := Field.FieldName; end;  TaDODBTest.GridColEnter(Sender TADODBTest.GridCole Taboutalist TaDODBTest.GridCole Taboutalist Tapodbar Tap		LogEvent('OnDataChange', Sender
StatusBar.Panels[0].Text := Field.ClassName; StatusBar.Panels[2].Text := NullStr[Field.IsNull]; TrackBlobs; if ActiveDataSet.UpdateStatus = usModified then ShowOriginalValues; end; end; end;  Connection Operations }  procedure TADODBTest.CheckConnection(CloseFirst: Boolean); const ConnectOptionValues: array [Boolean] of TConnectOption = (coConnectUnspecified, coAsyncConnect); var ConnStr: string; Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.ConnectonString:= ConnStr: = ConnectionString:= ConnStr: Connection.ConnectOptions := Connection.ConnectOptions :=  end; TADODBTest.GridColEnter(Sender ShullStr: array[Boolean] of string = (",','NLLI_'); var Field: TField;  procedure TrackBlobs; begin if Field.DataSet <> MasterDataSet then Exit; if (Field is TMemoField) and (Field > DBMemo1.DataField := Field.FieldName else if (Field > DBImage1.DataField := Field.FieldName; else if (Field > DBImage1.DataField := Field.FieldName; else if (Field > DBImage1.DataField := Field.FieldName; end;  'Yell All All All All All All All All All		
Field.ClassName;   StatusBar.Panels[2].Text :=   NullStr[Field.IsNull];   TADODBTest.GridColEnter(Sender : TObject);   Const   ShowOriginalValues;   end;   end;   end;   end;     Eld.Trield;	,	end;
NullStr[Field.IsNull];  TrackBlobs; if ActiveDataSet.UpdateStatus = usModified then  ShowOriginalValues; end; end; end;  { Connection Operations }  procedure  TADODBTest.CheckConnection(CloseFirst: Boolean); const  ConnectOptionValues: array [Boolean] of TConnectOption = (coConnectUnspecified, coAsyncConnect); var  ConnStr: string; Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.Connected then Exit; Connection.Close; MasterDataSource.DataSet then Exit; if (Field is TMemoField) and (Field ◇ DBMemo1.DataField := Field.FieldName else if (Field is TGraphicField) and (Field ◇ DBImage1.Field) then  DBImage1.DataField := Field.FieldName; end;  if not CloseFirst and Connection.Connected then Exit; Connection.Close; MasterClientData.Close;  begin if Field.Dataset.CanModify then begin V:= Field.OldValue;		
NullStr[Field.IsNull];  TrackBlobs; if ActiveDataSet.UpdateStatus = usModified then  ShowOriginalValues; end; end; end;  { Connection Operations }  procedure  TADODBTest.CheckConnection(CloseFirst: Boolean); const  ConnectOptionValues: array [Boolean] of TConnectOption = (coConnectUnspecified, coAsyncConnect); var  ConnStr: string; Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.Connected then Exit; Connection.Close; MasterDataSource.DataSet then Exit; if (Field is TMemoField) and (Field ◇ DBMemo1.DataField := Field.FieldName else if (Field is TGraphicField) and (Field ◇ DBImage1.Field) then  DBImage1.DataField := Field.FieldName; end;  if not CloseFirst and Connection.Connected then Exit; Connection.Close; MasterClientData.Close;  begin if Field.Dataset.CanModify then begin V:= Field.OldValue;	StatusBar.Panels[2].Text :=	procedure
TrackBlobs;  if ActiveDataSet.UpdateStatus = usModified then  ShowOriginalValues; end; end;  { Connection Operations }  procedure TADODBTest.CheckConnection(CloseFirst: Boolean); const  ConnectOptionValues: array [Boolean] of TConnectOption = (coConnectUnspecified, coAsyncConnect); var  ConnStr: string; Index: Integer; begin  if not CloseFirst and Connection.Connected then Exit; Connection.Connected then Exit; Connection.ConnectionString.Text; Connection.ConnectionString.Text; Connection.ConnectOptions :=  Connection.ConnectOptions :=  ShowOriginalValues; var  Index: Tobject); const  NullStr: array[Boolean] of string = (",', NULL '); var  Procedure TrackBlobs; begin  if Field.DataSet ⇔ MasterDataSource.DataSet then Exit; if (Field is TMemoField) and (Field ⇔ DBMemo1.Field) then DBMemo1.DataField := Field.FieldName else if (Field is TGraphicField) and (Field ⇔ DBImage1.Field) then DBImage1.DataField := Field.FieldName; end;  V: Variant; begin if Field.Dataset.CanModify then begin V:= Field.OldValue;		•
if ActiveDataSet.UpdateStatus = usModified then ShowOriginalValues; end; end;  {Connection Operations }  procedure TADODBTest.CheckConnection(CloseFirst: Boolean); const  ConnectOptionValues: array [Boolean] of TConnectOption = (coConnectUnspecified, coAsyncConnect); var  ConnStr: string; Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.Connected then Exit; Connection.ConnectionString.Text; Connection.ConnectionString.Text; Connection.ConnectionString.Text; Connection.ConnectOptions :=  const NullStr: array[Boolean] of string = (",'[NULL]'); var Field: TField;  procedure TrackBlobs; begin if Field.DataSet ⇔ MasterDataSource.DataSet then Exit; if (Field is TMemoField) and (Field ⇔ DBMemo1.Field) then DBImage1.DataField := Field.FieldName; else if (Field is TGraphicField) and (Field ⇔ DBImage1.Field) then DBImage1.DataField := Field.FieldName; else if (Field sat Source.DataSet then Exit; if (Field is TMemoField) and (Field ⇔ DBImage1.Field) then DBImage1.DataField := Field.FieldName; end; if rield.DataSet.CanModify then begin if Field.Dataset.CanModify then begin V := Field.OldValue;	-	· ·
usModified then ShowOriginalValues; end; end;  { Connection Operations }  procedure TADODBTest.CheckConnection(CloseFirst: Boolean); const ConnectOptionValues: array [Boolean] of TConnectOption = (coConnectUnspecified, coAsyncConnect); var ConnStr: string; Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.Connected then Exit; Connection.ConnectonString.Text; Connection.ConnectionString:= Connection.ConnectOptions:=  NullStr: array[Boolean] of string = (",',' NULL '); var Field: TField;  MasterDataSet ⇔ MasterDataSet then Exit; if (Field is TMemoField) and (Field ⇔ DBMemo1.DataField) := Field.FieldName else if (Field is TGraphicField) and (Field ⇔ DBImage1.Field) then DBImage1.DataField := Field.FieldName; end; if not CloseFirst and Connection.Connected then Exit; Connection.Connectof then Exit; Connection.Connectory := ConnectionString.Text; Connection.ConnectionString := Connection.ConnectOptions :=  V: Variant; begin if Field.Dataset.CanModify then begin V := Field.OldValue;		<b>Y</b> / /
ShowOriginalValues; end; end;  { Connection Operations }  procedure TADODBTest.CheckConnection(Clo seFirst: Boolean); const  ConnectOptionValues: array [Boolean] of TConnectOption = (coConnectUnspecified, coAsyncConnect); var  ConnStr: string; Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.Connected then Exit; Connection.Connectorstring.Text; Connection.ConnectionString.Text; Connection.ConnectOptions :=  ('','[NULL]'); var  Field: TField;  procedure TrackBlobs; begin if Field.DataSet ◇ MasterDataSource.DataSet then Exit; if (Field is TMemoField) and (Field ◇ DBMemo1.Field) then DBMemo1.DataField := Field.FieldName else if (Field is TGraphicField) then DBImage1.DataField := Field.FieldName; end;  var  V: Variant; begin if Field.Dataset.CanModify then begin V := Field.OldValue;	· ·	NullStr: array[Boolean] of string =
end; end; end;  { Connection Operations }  procedure TADODBTest.CheckConnection(CloseFirst: Boolean); const  ConnectOptionValues: array [Boolean] of TConnectOption = (coConnectUnspecified, coAsyncConnect); var  ConnStr: string; Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.Close; MasterClientData.Close; Connection.ConnectionString := Connection.ConnectionStrirg := Connection.ConnectOptions :=  var  procedure TrackBlobs; begin if Field.DataSet ⟨→ MasterDataSet then Exit; if (Field is TMemoField) and (Field ⟨→ DBMemo1.Field) then DBMemo1.DataField := Field.FieldName else if (Field is TGraphicField) and (Field ⟨→ DBImage1.DataField) := Field.FieldName; end; if not CloseFirst and Connection.Connected then Exit; Connection.Connector then Exit; begin if Field.Dataset.CanModify then begin V := Field.OldValue;		
Field: TField;	· · · · · · · · · · · · · · · · · · ·	
{ Connection Operations }  procedure TADODBTest.CheckConnection(CloseFirst: Boolean);  const  ConnectOptionValues: array [Boolean] of TConnectOption = (coConnectUnspecified, coAsyncConnect);  var  ConnStr: string; Index: Integer; begin  if not CloseFirst and Connection.Connected then Exit;  Connection.Connected then Exit;  Connection.ConnectionString.Text;  Connection.ConnectionString:= ConnStr;  Connection.ConnectOptions:=  procedure TrackBlobs; begin  if Field.DataSet <> MasterDataSource.DataSet then Exit;  if (Field is TMemoField) and (Field <> DBMemo1.Field) then  DBMemo1.DataField := Field.FieldName else if (Field is TGraphicField) and (Field <> DBImage1.DataField := Field.FieldName; end;  if not CloseFirst and Connection.Connected then Exit;  Connection.Connected then Exit;  Connection.Connectoonstring:= Connection.ConnectionString:= Connection.ConnectionString:= Connection.ConnectOptions:=		
begin  procedure  TADODBTest.CheckConnection(CloseFirst: Boolean);  const  ConnectOptionValues: array [Boolean] of TConnectOption = (coConnectUnspecified, coAsyncConnect);  var  ConnStr: string;  Index: Integer;  begin  if Field.DataSet ⇔  MasterDataSource.DataSet then  Exit;  if (Field is TMemoField) and (Field ⇔ DBMemo1.Field) then  DBMemo1.DataField := Field.FieldName  else if (Field is TGraphicField) and (Field ⇔ DBImage1.Field) then  DBImage1.DataField := Field.FieldName;  end;  if not CloseFirst and  Connection.Connected then Exit;  Connection.Connected then Exit;  Connection.Connected then Exit;  Connection.ConnectionString.Text;  Connection.ConnectionString := Connection.ConnectOptions := V := Field.OldValue;	Citu,	1 lotar 11 lotar
begin  procedure  TADODBTest.CheckConnection(CloseFirst: Boolean);  const  ConnectOptionValues: array [Boolean] of TConnectOption = (coConnectUnspecified, coAsyncConnect);  var  ConnStr: string;  Index: Integer;  begin  if Field.DataSet ⇔  MasterDataSource.DataSet then  Exit;  if (Field is TMemoField) and (Field ⇔ DBMemo1.Field) then  DBMemo1.DataField := Field.FieldName  else if (Field is TGraphicField) and (Field ⇔ DBImage1.Field) then  DBImage1.DataField := Field.FieldName;  end;  if not CloseFirst and  Connection.Connected then Exit;  Connection.Connected then Exit;  Connection.Connected then Exit;  Connection.ConnectionString.Text;  Connection.ConnectionString := Connection.ConnectOptions := V := Field.OldValue;	{ Connection Operations }	procedure TrackBlobs:
if Field.DataSet ⇔  MasterDataSource.DataSet then Exit;  const  ConnectOptionValues: array [Boolean] of TConnectOption = (coConnectUnspecified, coAsyncConnect);  var  ConnStr: string;  Index: Integer; begin if not CloseFirst and Connection.Connected then Exit;  Connection.Connected then Exit;  Connection.ConnectionString.Text;  Connection.ConnectionString.Text;  Connection.ConnectOptions :=  if Field.DataSet ⇔ MasterDataSource.DataSet then Exit;  if (Field is TMemoField) and (Field ⇔ DBMemo1.Field) then  DBMemo1.DataField := Field.FieldName else if (Field is TGraphicField) and (Field ⇔ DBImage1.Field) then  DBImage1.DataField := Field.FieldName; end;  var  V: Variant; begin  if Field.Dataset.CanModify then begin  V := Field.OldValue;	( Connection Operations )	
TADODBTest.CheckConnection(Clo seFirst: Boolean); const  ConnectOptionValues: array [Boolean] of TConnectOption = (coConnectUnspecified, coAsyncConnect); var  ConnStr: string; Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.Close; MasterDataSource.DataSet then Exit; if (Field is TMemoField) and (Field <> DBMemo1.DataField := Field.FieldName else if (Field is TGraphicField) and (Field <> DBImage1.Field) then DBImage1.DataField := Field.FieldName; end;  procedure ShowOriginalValues; var V: Variant; begin if Field.Dataset.CanModify then begin V:= Field.OldValue;	procedure	
seFirst: Boolean);  const  ConnectOptionValues: array [Boolean] of TConnectOption = (coConnectUnspecified, coAsyncConnect);  var  ConnStr: string; Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.Close; MasterClientData.Close; ConnStr: = ConnectionString.Text; Connection.ConnectOptions :=  ConnStr;  Connection.ConnectOptions :=  Exit; if (Field is TMemoField) and (Field <> DBMemo1.DataField := Field.FieldName else if (Field is TGraphicField) and (Field <> DBImage1.DataField := Field.FieldName; end;  if not CloseFirst and Connection.Connected then Exit; Connection.Connected then Exit;  Connection.Connectooptions :=  Exit;  if (Field is TMemoField) and (Field <> DBMemo1.Field) then DBImage1.DataField := Field.FieldName; end;  if Field.Shame else if (Field is TMemoField) and (Field <> DBMemo1.Field) then DBImage1.DataField := Field.FieldName; end;  if Field.DataField := Field.FieldName; end;  if Field.DataField := Field.FieldName; end;  if Field.FieldName;		
ConnectOptionValues: array   (Field is TMemoField) and     ConnectOptionValues: array   (Field ⇔ DBMemo1.Field) then     CoConnectUnspecified, coAsyncConnect); var   Else if (Field is TGraphicField)     Var	· ·	
ConnectOptionValues: array   (Field  DBMemo1.Field) then     CoConnectUnspecified, coAsyncConnect);   Var   ConnStr: string;   Index: Integer;   begin   if not CloseFirst and   Connection.Connected then Exit;   Connection.Close;   MasterClientData.Close;   ConnStr: = ConnectionString.Text;   Connection.ConnectionString: = ConnStr;   Connection.ConnectOptions :=   Connection.ConnectOptions :=   V: = Field.OldValue;   V: = Field.OldValue;	,	
Boolean   of TConnectOption = (coConnectUnspecified, coAsyncConnect);		
(coConnectUnspecified, coAsyncConnect);  var  ConnStr: string; Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.Close; MasterClientData.Close; ConnStr: = ConnectionString.Text; Connection.ConnectOptions :=  Connection.ConnectOptions :=  Field.FieldName else if (Field is TGraphicField) and (Field ◇ DBImage1.Field) then  DBImage1.DataField := Field.FieldName; else if (Field is TGraphicField) and (Field ◇ DBImage1.Field) then  DBImage1.DataField := Field.FieldName; else if (Field is TGraphicField) and (Field ◇ DBImage1.Field) then  DBImage1.DataField := Field.FieldName; else if (Field is TGraphicField) and (Field ◇ DBImage1.Field) then  DBImage1.DataField := Field.FieldName; else if (Field is TGraphicField) and (Field ◇ DBImage1.Field) then  DBImage1.DataField := Field.FieldName; else if (Field is TGraphicField) and (Field ◇ DBImage1.Field) then  DBImage1.DataField := Field.FieldName  V: Variant; begin  if Field.Dataset.CanModify then  begin  V := Field.OldValue;	_	
coAsyncConnect);  var  ConnStr: string;  Index: Integer;  begin  if not CloseFirst and Connection.Connected then Exit;  Connection.Close;  MasterClientData.Close;  ConnStr := ConnectionString.Text;  Connection.ConnectOptions :=  Connection.ConnectOptions :=  else if (Field is TGraphicField) and (Field <> DBImage1.DataField := Field.FieldName; end;  procedure ShowOriginalValues; var  V: Variant; begin  if Field.Dataset.CanModify then begin  V := Field.OldValue;	1	
var  ConnStr: string;  Index: Integer;  begin  if not CloseFirst and Connection.Connected then Exit;  Connection.Close;  MasterClientData.Close;  ConnStr := ConnectionString.Text;  Connection.ConnectionString :=  ConnStr;  Connection.ConnectOptions :=  and (Field ❖ DBImage1.Field) then  DBImage1.DataField :=  Field.FieldName;  end;  procedure ShowOriginalValues;  var  V: Variant;  begin  if Field.Dataset.CanModify then  begin  V := Field.OldValue;		
ConnStr: string; Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.Close; MasterClientData.Close; ConnStr := ConnectionString.Text; Connection.ConnectionString := ConnStr; Connection.ConnectOptions :=  DBImage1.DataField := Field.FieldName; end;  var  procedure ShowOriginalValues;  V: Variant; begin if Field.Dataset.CanModify then begin  V := Field.OldValue;	, ,	1
Index: Integer; begin if not CloseFirst and Connection.Connected then Exit; Connection.Close; MasterClientData.Close; ConnStr := ConnectionString.Text; Connection.ConnectionString := ConnStr; Connection.ConnectOptions :=  Field.FieldName; end;  Procedure ShowOriginalValues;  V: Variant; begin if Field.Dataset.CanModify then begin  V := Field.OldValue;		
begin  if not CloseFirst and Connection.Connected then Exit; Connection.Close;  MasterClientData.Close; ConnStr := ConnectionString.Text; Connection.ConnectionString := ConnStr;  Connection.ConnectOptions :=  end;  procedure ShowOriginalValues;  var  V: Variant; begin  if Field.Dataset.CanModify then begin  V := Field.OldValue;		8
if not CloseFirst and Connection.Connected then Exit; Connection.Close;  MasterClientData.Close; ConnStr := ConnectionString.Text; Connection.ConnectionString := ConnStr;  Connection.ConnectOptions :=  if Field.Dataset.CanModify then begin  V := Field.OldValue;		
Connection.Connected then Exit; Connection.Close;  MasterClientData.Close; ConnStr := ConnectionString.Text; Connection.ConnectionString := ConnStr; Connection.ConnectOptions :=  procedure ShowOriginalValues;  var  V: Variant; begin if Field.Dataset.CanModify then begin  V := Field.OldValue;	B	enu,
Connection.Close;  MasterClientData.Close;  ConnStr := ConnectionString.Text;  Connection.ConnectionString :=  ConnStr;  Connection.ConnectOptions :=  var  V: Variant;  begin  if Field.Dataset.CanModify then  begin  V := Field.OldValue;		nrogoduro ChowOwiginalValuas
MasterClientData.Close;V: Variant;ConnStr := ConnectionString.Text;beginConnection.ConnectionString :=if Field.Dataset.CanModify thenConnection.ConnectOptions :=V := Field.OldValue;	·	
ConnStr := ConnectionString.Text; Connection.ConnectionString := ConnStr; Connection.ConnectOptions :=  begin if Field.Dataset.CanModify then begin V := Field.OldValue;	,	
Connection.ConnectionString := if Field.Dataset.CanModify then begin  Connection.ConnectOptions := V := Field.OldValue;	·	
ConnStr; begin Connection.ConnectOptions := V := Field.OldValue;		
Connection.ConnectOptions := V := Field.OldValue;	_	-
	ŕ	
ConnectOptionValues[AsyncConnec if not VarIsNull(V) and (V <>	_	
	ConnectOptionValues[AsyncConnec	if not VarIsNull(V) and (V <>

	t.Checked];
procedure	if UseShapeProvider.Checked then
TADODBTest.ConnectionStringClic	Connection.Provider :=
k(Sender: TObject);	'MSDataShape';
begin	Index :=
if (ConnectionString.Text <> '')	ConnectionString.Items.IndexOf(Co
and not	nnStr);
ConnectionString.DroppedDown	if Index > 0 then
then	
begin	ConnectionString.Items.Delete(Inde
CheckConnection(True);	x);
MasterTableName.Items.Clear;	if Index <> 0 then
MasterTableName.Text := ";	begin
DetailTableName.Items.Clear;	ConnectionString.Items.Insert(0,
DetailTableName.Text := '';	ConnStr);
MasterProcName.Items.Clear;	while
MasterProcName.Text := ";	ConnectionString.Items.Count > 20
DetailProcName.Items.Clear;	do
DetailProcName.Text := ";	
end;	ConnectionString.Items.Delete(20);
end;	end;
	ConnectionString.ItemIndex := 0;
procedure	Application.ProcessMessages;
TADODBTest.ConnectionStringKey	Connection.Open;
Press(Sender: TObject; var Key:	if AsyncConnect.Checked and
Char);	(stConnecting in Connection.State)
begin	then
if Key = #13 then	ShowProgressBar('Connecting');
begin	//
if	ShowProperties(Connection.Propert
ConnectionString.DroppedDown	ies);
then	end;
ConnectionString.DroppedDown	
:= False;	procedure
ConnectionStringClick(Sender);	TADODBTest.EditConnStrClick(Se
Key := #0;	nder: TObject);
end;	begin
end;	Connection.Close;
	Connection.ConnectionString :=
procedure	ConnectionString.Text;
TADODBTest.CloseConnectionExec	if
ute(Sender: TObject);	EditConnectionString(Connection)
begin	then
Connection.Close;	begin
end;	ConnectionString.Text :=
	Connection.ConnectionString;
{ Common DataSet Operations }	ConnectionStringClick(Sender);
	end;
	end;

ADOSource.Connection := nil;	procedure
end;	TADODBTest.OpenDataSet(Source:
,	TCustomADODataSet);
function	
TADODBTest.GetActiveDataSet:	procedure ShowFetchProgress;
TDataSet;	begin
begin	while stFetching in
if not Assigned(FActiveDataSet)	ADOSource.RecordSetState do
then	begin
DatabaseError('No active	with ADOSource do
dataset');	StatusBar.Panels[3].Text :=
Result := FActiveDataSet;	Format('%d of %d', [RecNo,
end;	RecordCount]);
,	Application.ProcessMessages;
procedure	end;
TADODBTest.SetActiveDataSet(Val	end;
ue: TDataSet);	
	begin
function GetDetailDataSet:	ClearEventLog.Execute;
TDataSet;	Screen.Cursor := crHourGlass;
var	try
I: Integer;	Source.Close;
begin	MasterClientData.Close;
Result := nil;	ADOSource :=
if (Value = MasterTable) and	TCustomADODataSet(Source);
DetailTable.Active then	
Result := DetailTable	SetRecordSetEvents(UseClientCurso
else if (Value = MasterQuery) and	r.Checked, ADOSource);
DetailQuery.Active then	Provider.DataSet := ADOSource;
Result := DetailQuery	if MidasButton.Checked then
else if (Value = MasterProc) and	ActiveDataSet := MasterClientData
DetailProc.Active then	else
Result := DetailProc	begin
else for I := 0 to	ActiveDataSet := Source;
Value.Fields.Count - 1 do	ShowFetchProgress;
if Value.Fields[I] is	end;
TDataSetField then	if MasterGrid.Visible then
begin	MasterGrid.SetFocus;
Result :=	finally
TDataSetField(Value.Fields[I]).Nest	Screen.Cursor := crDefault;
edDataSet;	end;
Break;	StreamSettings(True);
end;	end;
end;	
	procedure
begin	TADODBTest.DisconnectDataSetEx
StatusBar.Panels[2].Text := ";	ecute(Sender: TObject);
MasterDataSource.Enabled :=	begin

if Connection = nil then	False;
Connection := Self.Connection;	<b>DetailDataSource.Enabled</b> := False;
CheckConnection(False);	try
if UseClientCursor.Checked then	MasterGrid.DataSource := nil;
CursorLocation := clUseClient	FActiveDataSet := Value;
else	DetailDataSource.DataSet := nil;
CursorLocation := clUseServer;	MasterDataSource.DataSet :=
for I := 0 to	Value;
CursorTypeItem.Count - 1 do	if Assigned(Value) then
if	begin
CursorTypeItem.Items[I].Checked	Value.Open;
then	if AsyncExecute.Checked and
begin	(Value.State = dsOpening) then
CursorType := TCursortype(I);	ShowProgressBar('Executing');
Break;	if DisplayDetails.Checked then
end;	DetailDataSource.DataSet :=
for I := 0 to LockTypeItem.Count	GetDetailDataSet;
- 1 do	end;
if	BindControls(Value);
LockTypeItem.Items[I].Checked	finally
then	MasterDataSource.Enabled :=
begin	True;
LockType := TADOLocktype(I);	DetailDataSource.Enabled :=
Break;	True;
end;	end;
ExecuteOptions := [];	if Assigned(Value) then
if AsyncExecute.Checked then	begin
ExecuteOptions :=	Update;
[eoAsyncExecute];	StatusMsg := 'ActiveDataSet is ' +
if AsyncFetch.Checked then	Value.Name;
ExecuteOptions :=	if
ExecuteOptions +	Assigned(AreaSelector.ActivePage.O
[eoAsyncFetchNonBlocking];	nShow) then
MaxRecords :=	,
Self.MaxRecords.Tag;	AreaSelector.ActivePage.OnShow(ni
EnableBCD :=	l);
Self.EnableBCD.Checked;	end;
end;	end;
LogEvent('BeforeOpen', DataSet);	
end;	procedure
	TADODBTest.DataSetBeforeOpen(
procedure	DataSet: TDataSet);
TADODBTest.DataSetAfterOpen(D	var
ataSet: TDataSet);	I: Integer;
begin	begin
ClearProgressBar;	with DataSet as
//	TCustomADODataSet do
ShowProperties(ADOSource.Record	begin

end;	Set.Fields[0].Properties);
end;	LogEvent('AfterOpen', DataSet);
	end;
procedure	ciru,
TADODBTest.MasterTableNameKe	procedure
yPress(Sender: TObject; var Key:	TADODBTest.DataSetAfterClose(D
Char);	ataSet: TDataSet);
begin	begin
if Key = #13 then	LogEvent('AfterClose', DataSet);
begin	if DataSet = FActiveDataSet then
with Sender as TComboBox do	
	FActiveDataSet := nil;
if DroppedDown then DroppedDown := False;	if DataSet = ADOSource then
OpenTable.Execute;	FilterGroupBox.ItemIndex := -1;
•	end;
Key := #0;	www.aadwwa
end;	procedure TAPOPPTost PotoSetFotah Complete
end;	TADODBTest.DataSetFetchComplet
nuo as duno	e(DataSet: TCustomADODataSet;
procedure	const Error: Error; var
TADODBTest.MasterTableNameCli	EventStatus: TEventStatus);
ck(Sender: TObject);	begin
begin	LogEvent('FetchComplete',
with Sender as TComboBox do	DataSet);
if not DroppedDown then	end;
begin	
DetailTableName.Text := ";	procedure
OpenTable.Execute;	TADODBTest.FormCloseQuery(Sen
end;	der: TObject;
end;	var CanClose: Boolean);
	begin
procedure	CloseActiveDataSet.Execute;
TADODBTest.DetailTableNameClic	end;
k(Sender: TObject);	
begin	{ Table Operations }
with Sender as TComboBox do	
if not DroppedDown and	procedure
(DetailTable.TableName <> Text)	TADODBTest.MasterTableNameDr
then	opDown(Sender: TObject);
OpenTable.Execute;	begin
end;	try
	CheckConnection(False);
procedure	with Sender as TComboBox do
TADODBTest.UpdateReOpenMenu;	if Items.Count < 1 then
var	
I: Integer;	Connection.GetTableNames(Items);
begin	except
while FileReOpen.Count > 0 do	{ Eat any exceptions so the
FileReOpen.Items[0].Free;	combobox doesn't paint funny }
1 [.] 7	

	for I := 0 to FClosedTables.Count -
nyoooduyo	1 do
procedure TADODBTest.CloseActiveDataSetE	1 40
	ElaDaOn an Add(Navyltann(Farmant)
xecute(Sender: TObject);	FileReOpen.Add(NewItem(Format('
begin	%d) %s', [I, FClosedTables[I]]), 0,
ActiveDataSet.Close;	False,
end;	True, ClosedFileClick, 0, ''));
	end;
procedure	
TADODBTest.MasterTableBeforeO	procedure
pen(DataSet: TDataSet);	TADODBTest.ClosedFileClick(Send
begin	er: TObject);
DataSetBeforeOpen(DataSet);	var
MasterTable.TableDirect :=	S: string;
UseTableDirect.Checked;	Index, P, P2: Integer;
MasterTable.TableName :=	begin
MasterTableName.Text;	S :=
DetailTable.MasterSource := nil;	Copy(TMenuItem(Sender).Caption,
end;	5, MAXINT);
	P := Pos(':', S);
procedure	P2 := Pos('/', S);
TADODBTest.DetailTableBeforeOp	if P2 > 0 then
en(DataSet: TDataSet);	DetailTableName.Text := Copy(S,
begin	P2+1, MAXINT) else
DataSetBeforeOpen(DataSet);	begin
DetailTable.TableDirect :=	DetailTableName.Text := '';
UseTableDirect.Checked;	<b>P2</b> := <b>MAXINT</b> ;
DetailTable.TableName :=	end;
DetailTableName.Text;	MasterTableName.Text := Copy(S,
end;	P+1, P2-P-1);
	Index :=
procedure	FClosedTables.IndexOf(S);
TADODBTest.MasterTableAfterOp	if Index > -1 then
en(DataSet: TDataSet);	FClosedTables.Delete(Index);
var	OpenTable.Execute;
I: Integer;	end;
Field: TField;	cnu,
MasterFields: string;	procedure
begin	TADODBTest.OpenTableExecute(S
	ender: TObject);
if DetailTableName.Text <> " then	<b>v</b> //
begin DeteilTable Onem	begin if MasterTableName.Text <> ''
DetailTable.Open;	
if MasterTableName.Text =	then
DetailTableName.Text then	begin
MasterFields :=	MasterTable.Close;
MasterTable.Fields[0].FieldName+';	OpenDataSet(MasterTable);
,	end;
else	end;

MasterTable.TableName;	for $I := 0$ to
if DetailTable.Active then	DetailTable.Fields.Count - 1 do
begin Tabla Factors   1/1	begin Field :=
TableEntry := TableEntry + '/' +	
DetailTable.TableName;	MasterTable.FindField(DetailTable.
DetailTable.Close;	Fields[I].FieldName);
end;	if Field <> nil then
if	begin
FClosedTables.IndexOf(TableEntry)	if
= -1 then	DetailTable.IndexDefs.GetIndexFor
begin	Fields(MasterFields +
FClosedTables.Insert(0,	Field.FieldName, False) <> nil then
TableEntry);	MasterFields := MasterFields
if FClosedTables.Count > 9 then	+ Field.FieldName + ';';
FClosedTables.Delete(9);	end;
end;	end;
end;	if MasterFields = " then
	DatabaseError('Cannot
begin	determine linking fields for detail');
UpdateClosedTables;	SetLength(MasterFields,
DetailTable.Close;	Length(MasterFields)-1);
DataSetBeforeClose(Dataset);	DetailTable.IndexFieldNames :=
end;	MasterFields;
- Chu,	DetailTable.MasterFields :=
{ Query Operations }	MasterFields;
(Query operations)	DetailTable.MasterSource :=
procedure	DetailMasterSource;
TADODBTest.ProcessQuery(Select	end;
Query: Boolean);	DataSetAfterOpen(DataSet);
Query: Booteury;	end;
procedure UpdateQueryHistory;	,
var	procedure
DSQL: string;	TADODBTest.FileMenuClick(Sende
	r: TObject);
begin if	begin
FMasterQueries.IndexOf(MasterSQ	UpdateReOpenMenu;
L.Text) <> -1 then Exit;	FileReOpen.Enabled :=
L. Text) > -1 then Exit,	FClosedTables.Count > 0;
EMastarQuarias Add/MastarSQLT	end;
FMasterQueries.Add(MasterSQL.T	
ext);	procedure
DSQL := DetailSQL.Text;	TADODBTest.MasterTableBeforeCl
if DSQL = " then DSQL :=	ose(DataSet: TDataSet);
'(empty)';	ose(Damoen i Damoet);
FDetailQueries.Insert(0, DSQL);	procedure UpdateClosedTables;
if FMasterQueries.Count > 9 then	•
begin D. L. (2)	Var TableEntry: string:
FMasterQueries.Delete(0);	TableEntry: string;
FDetailQueries.Delete(0);	begin TableFatava -
	TableEntry :=

end;	end;
cnu,	end;
nyoooduyo	enu,
procedure TADODBTest.ExecSQLExecute(Sen	VOM
der: TObject);	Paganda Affacta da Intagana
<b>9</b> / ·	RecordsAffected: Integer;
begin  Process On over (Folso):	begin
ProcessQuery(False);	CheckConnection(False);
end;	if SelectQuery then
,	begin
procedure	MasterQuery.Close;
TADODBTest.OpenQueryExecute(S	MasterQuery.SQL.Text :=
ender: TObject);	MasterSQL.Text;
begin	WriteParameterData;
ProcessQuery(True);	OpenDataSet(MasterQuery)
end;	end else
	begin
procedure	if SQLParams.Checked then
TADODBTest.PrevQueryUpdate(Se	begin
nder: TObject);	ADOCommand.CommandType
begin	:= cmdText;
PrevQuery.Enabled :=	ADOCommand.CommandText
FQueryIndex <	:= MasterSQL.Text;
(FMasterQueries.Count - 1);	end else
end;	begin
	ADOCommand.CommandType
procedure	:= cmdStoredProc;
TADODBTest.PrevQueryExecute(Se	ADOCommand.CommandText
nder: TObject);	:= MasterProcName.Text;
begin	end;
Assert(FQueryIndex <	if ParameterList.Items.Count > 0
(FMasterQueries.Count - 1));	then
Inc(FQueryIndex);	begin
SetQueryText;	WriteParameterData;
end;	,
	ADOCommand.Parameters.Assign(
procedure	FParamSource);
TADODBTest.NextQueryExecute(Se	end;
nder: TObject);	,
begin	ADOCommand.Execute(RecordsAff
if FQueryIndex > -1 then	ected, EmptyParam);
Dec(FQueryIndex);	StatusMsg := Format('%d rows
SetQueryText;	were affected', [RecordsAffected]);
end;	if ProcParams.Checked then
,	2 2 002 112 112 112 112 112 112 112 112
procedure	MasterProc.Parameters.Assign(AD
TADODBTest.MasterSQLKeyPress(	OCommand.Parameters);
Sender: TObject; var Key: Char);	end;
begin	UpdateQueryHistory;
Megili	opuatequelymstory,

MasterSQL.Text;	if Key = #13 then
end;	begin
enu;	
	OpenQuery.Execute;
procedure	Key := #0;
TADODBTest.DetailQueryBeforeOp	end;
en(DataSet: TDataSet);	end;
begin	
DataSetBeforeOpen(DataSet);	procedure
DetailQuery.SQL.Text :=	TADODBTest.SetQueryText;
DetailSQL.Text;	var
DetailQuerySource.Dataset :=	DSQL: string;
MasterQuery;	begin
if DetailQuery.Parameters.Count =	if FQueryIndex > -1 then
0 then	begin
	MasterSQL.Text :=
RefreshParameters(DetailQuery.Par	FMasterQueries[FQueryIndex];
ameters);	DSQL :=
end;	FDetailQueries[FQueryIndex];
	if DSQL = '(empty)' then DSQL :=
procedure	":
TADODBTest.MasterQueryAfterOp	DetailSQL.Text := DSQL;
en(DataSet: TDataSet);	end else
begin	begin
if Trim(DetailSQL.Text) <> " then	MasterSQL.Text := ";
DetailQuery.Open else	DetailSQL.Text := ";
DetailQuerySource.Dataset := nil;	end;
DataSetAfterOpen(DataSet);	end;
end;	chu,
	nuonaduma
procedure	procedure TADODBTest.EditCommandTextCl
TADODBTest.MasterQueryBeforeC	
lose(DataSet: TDataSet);	ick(Sender: TObject);
begin	Commands strings
DetailQuery.Close;	Command: string;
DataSetBeforeClose(DataSet);	begin
end;	CheckConnection(False);
ciiu,	Command := MasterSQL.Text;
(Stowed Dwoodywas)	if EditSQL(Command,
{ Stored Procedures }	Connection.GetTableNames,
	Connection.GetFieldNames) then
procedure	MasterSQL.Text := Command;
TADODBTest.MasterProcBeforeOp	end;
en(DataSet: TDataSet);	
begin	procedure
DataSetBeforeOpen(DataSet);	TADODBTest.MasterQueryBeforeO
MasterProc.ProcedureName :=	pen(DataSet: TDataSet);
MasterProcName.Text;	begin
WriteParameterData;	DataSetBeforeOpen(DataSet);
end;	MasterQuery.SQL.Text :=

var Key: Char);	
begin	procedure
if Key = #13 then	TADODBTest.MasterProcAfterOpe
begin	n(DataSet: TDataSet);
with Sender as TComboBox do	begin
	if DetailProcName.Text <> " then
if DroppedDown then	
DroppedDown := False;	DetailProc.Open;
Key := #0;	DataSetAfterOpen(DataSet);
end;	end;
	,
end;	procedure
,	TADODBTest.DetailProcBeforeOpe
procedure	n(DataSet: TDataSet);
TADODBTest.MasterProcNameClic	begin
k(Sender: TObject);	DataSetBeforeOpen(DataSet);
begin	DetailProc.ProcedureName :=
with Sender as TComboBox do	DetailProcName.Text;
if not DroppedDown then	
DetailProcName.Text := ";	RefreshParameters(DetailProc.Para
end;	meters);
	end;
procedure	
TADODBTest.DetailProcNameClick	procedure
(Sender: TObject);	TADODBTest.OpenProcedureExecu
begin	te(Sender: TObject);
end;	begin
	if MasterProcName.Text <> " then
{ Packet Save/Load }	begin
	MasterProc.Close;
procedure	OpenDataSet(MasterProc);
TADODBTest.LoadFromFileExecut	end;
e(Sender: TObject);	end;
begin	
OpenDialog.FilterIndex := 1;	procedure
OpenDialog.FileName :=	TADODBTest.ProcNameDropDown
FLastDataFile;	(Sender: TObject);
if OpenDialog.Execute then	begin
begin	CheckConnection(False);
0	with Sender as TComboBox do
ADODataSet.LoadFromFile(OpenDi	if Items.Count < 1 then
alog.FileName);	
FLastDataFile :=	Connection.GetProcedureNames(Ite
OpenDialog.FileName;	ms);
ActiveDataSet := ADODataSet;	end;
ADOSource := ADODataSet;	
end;	procedure
end;	TADODBTest.MasterProcNameKey
cnu,	Press(Sender: TObject;
	i i essessimet. I Object,

1.	
end;	procedure
	TADODBTest.SaveToFileExecute(S
procedure	ender: TObject);
TADODBTest.StreamFormInClick(	begin
Sender: TObject);	SaveDialog.FilterIndex := 1;
var	SaveDialog.FileName :=
Form: TADODBTest;	FLastDataFile;
begin	if SaveDialog.Execute then
OpenDialog.FilterIndex := 2;	begin
OpenDialog.FileName :=	
FLastFormFile;	ADOSource.SaveToFile(SaveDialog.
if OpenDialog.Execute then	FileName, pfADTG);
begin	FLastDataFile :=
Form :=	SaveDialog.FileName;
TADODBTest.CreateNew(Applicati	end;
on, 0);	end;
	,
ReadComponentResFile(OpenDialo	procedure
g.FileName, Form);	TADODBTest.FileActionsUpdate(Se
FLastFormFile :=	nder: TObject);
OpenDialog.FileName;	begin
Form.FormCreate(Form);	SaveToFile.Enabled :=
end;	Assigned(FActiveDataSet) and
end;	ActiveDataSet.Active;
enu,	CloseActiveDataSet.Enabled :=
(DD Control Linking)	SaveToFile.Enabled;
{ DB Control Linking }	DisconnectDataset.Enabled :=
	SaveToFile.Enabled;
procedure	CloseConnection.Enabled :=
TADODBTest.BindControls(DataSe	Connection.Connected;
t: TDataSet);	,
I DI DEFENC	end;
procedure DeleteDBEditControls;	(Sangaring)
var	{ Streaming }
I: Integer;	
begin	procedure
with DBEditScroller do	TADODBTest.StreamFormOutClick
for I := ComponentCount - 1	(Sender: TObject);
downto 0 do	begin
if (Components[I] is TDBEdit)	SaveDialog.FilterIndex := 2;
or (Components[I] is TLabel) then	SaveDialog.FileName :=
Components[I].Free;	FLastFormFile;
end;	if SaveDialog.Execute then
	begin
procedure	
SetDisplayType(ForwardOnly:	WriteComponentResFile(SaveDialog
Boolean);	.FileName, Self);
begin	FLastFormFile :=
if ForwardOnly then	SaveDialog.FileName;
	end;

Γ -	1
begin	b
AutoSize := False;	
Alignment := taRightJustify;	
Left := 3;	
Top := LabelTop;	Gri
Width := 59;	
Parent := DBEditScroller;	DB
Caption := F.DisplayLabel+':';	on:
end;	
end;	DB
	on:
var	
I: Integer;	[nb]
Field: TField;	
begin	e
DBMemo1.DataField := '';	b
DBImage1.DataField := '';	
DetailGrid.Visible := False;	
GridSplitter.Visible := False;	Mas
BlobCtrlPanel.Visible := False;	
DBImage1.Visible := False;	
DBMemo1.Visible := False;	[nb]
ReadOnlyLabel.Visible := False;	
if Assigned(DataSet) and	nbP
DataSet.Active then	e
begin	en
SetDisplayType((DataSet is	
TCustomADODataSet) and not	pr
Teustom/Abobataset) and not	TFi
TCustomADODataSet(DataSet).Sup	va
ports([coBookmark,	L
coMovePrevious]));	be
for I := 0 to DataSet.FieldCount -	W
1 do	TD
begin	b
Field := DataSet.Fields[I];	
Field.OnValidate :=	
FieldValidate;	+ 5)
case Field.DataType of	
ftMemo:	
if DBMemo1.DataField = ''	Car
then	
begin	
DBMemo1.DataField :=	Mas
Field.FieldName;	
DBMemo1.Visible := True;	e
· · · · · · · · · · · · · · · · · · ·	W
end;	TL
ftGraphic:	]

begin
MasterGrid.Visible := False;
MasterGrid.DataSource := nil;
DBEditScroller.Height :=
GridPanel.Height;
DBEditScroller.HorzScrollBar.Positi on := 0;
DBEditScroller.VertScrollBar.Positi on := 0;
DBNavigator1.VisibleButtons :=
[nbNext, nbInsert, nbDelete, nbEdit,
nbPost, nbCancel, nbRefresh];
end else
begin
MasterGrid.Visible := True;
MasterGrid.DataSource :=
MasterDataSource;
DBEditScroller.Height := 0;
DBNavigator1.VisibleButtons :=
[nbFirst, nbPrior, nbNext, nbLast,
nbInsert, nbDelete, nbEdit,
nbPost, nbCancel, nbRefresh];
end;
end;
procedure CreateDBEdit(F:
TField);
var
LabelTop: Integer;
begin
with
TDBEdit.Create(DBEditScroller) do
begin
Left := 65;
Top := (F.FieldNo - 1) * (Height + 5) + 5;
LabelTop := Top + 3;
Width := F.DisplayWidth *
Canvas.TextWidth('0');
Parent := DBEditScroller;
DataSource :=
MasterDataSource;
DataField := F.FieldName;
end;
with
TLabel.Create(DBEditScroller) do

: TObject);	if DBImage1.DataField = ''
begin	then
ě	begin
if not Assigned(FActiveDataSet) then Exit;	DBImage1.DataField :=
,	DataSet.Fields[I].FieldName;
ActiveDataSource := (Sender as	
TDBGrid).DataSource;	DBImage1.Visible := True;
DBNavigator1.DataSource :=	end;
ActiveDataSource;	ftDataSet, ftReference:
	if DisplayDetails.Checked and
DataSourceDataChange(ActiveData	(DetailDataSource.DataSet = nil)
Source, nil);	then
end;	begin
	DetailDataSource.DataSet :=
procedure	TDataSetField(DataSet.Fields[I]).Ne
TADODBTest.PopupMenu1Popup(S	stedDataSet;
ender: TObject);	end;
var	ftBytes, ftVarBytes:
I: Integer;	begin
MI: TMenuItem;	Field.OnGetText :=
F, CurField: TField;	BinaryGetText;
begin	Field.OnSetText :=
with PopupMenu1, ActiveDataSet	BinarySetText;
do	Field.DisplayWidth :=
begin	(Field.Size + 3);
if PopupMenu1.PopupComponent	end;
= DBMemo1 then	else
CurField := DBMemo1.Field else	if not MasterGrid.Visible then
CurField := DBImage1.Field;	CreateDBEdit(Field);
while Items.Count > 0 do	end;
Items.Delete(0);	end;
	,
MI := NewItem('(None)', 0, False,	BlobCtrlPanel.Visible :=
True, FieldSelect, 0, 'None');	DBMemo1.Visible or
Items.Add(MI);	DBImage1.Visible;
for I := 0 to FieldCount - 1 do	ReadOnlyLabel.Visible := not
if Fields[I] is TBlobField then	DataSet.CanModify;
begin	if
F := Fields[I];	Assigned(DetailDataSource.DataSet)
MI := NewItem(F.FieldName, 0,	then
F=CurField, True, FieldSelect, 0,	begin
'mi'+F.FieldName);	GridSplitter.Visible := True;
MI.Tag := Integer(F);	DetailGrid.Visible := True;
Items.Add(MI);	end;
end;	end else
end;	DeleteDBEditControls;
end;	end;
,	,
procedure	procedure
TADODBTest.FieldSelect(Sender:	TADODBTest.GridSetFocus(Sender
1112 ODD I COMI ICIADORCO DORIGO	1112 ODD I CON GITADON OCUSIOCITACI

MidasApplyUpdates.Enabled :=	TObject);
MasterClientData.Active and	var
((MasterClientData.ChangeCount	MI: TMenuItem;
> 0) or (MasterClientData.State in	begin
dsEditModes));	MI := TMenuItem(Sender);
MidasCancelUpdates.Enabled :=	if PopupMenu1.PopupComponent
MidasApplyUpdates.Enabled;	= DBImage1 then
LoadBlobFromFile.Enabled :=	try
Enabled and	if MI.Tag = 0 then
(MasterGrid.SelectedField is	DBImage1.DataField := " else
TBlobField);	DBImage1.DataField :=
end;	TField(MI.Tag).FieldName;
	except
procedure	DBImage1.DataField := '';
TADODBTest.BatchUpdateExecute(	raise;
Sender: TObject);	end
begin	else if
if ADOSource.Connection = nil	PopupMenu1.PopupComponent =
then	DBMemo1 then
begin	try
CheckConnection(False);	if MI.Tag = 0 then
ADOSource.Connection :=	DBMemo1.DataField := " else
Connection;	DBMemo1.DataField :=
end;	TField(MI.Tag).FieldName;
ADOSource.UpdateBatch;	except
end;	DBMemo1.DataField := '';
	raise;
procedure	end;
TADODBTest.CancelBatchExecute(	end;
Sender: TObject);	
begin	{ Editing / Updates }
ADOSource.CancelUpdates;	(======================================
end;	procedure
	TADODBTest.EditActionsUpdate(S
procedure	ender: TObject);
TADODBTest.ClearFieldExecute(Se	var
nder: TObject);	Enabled: Boolean;
var	begin
Field: TField;	Enabled :=
begin	Assigned(FActiveDataSet);
Field := MasterGrid.SelectedField;	BatchUpdate.Enabled :=
if Field = nil then Exit;	Assigned(ADOSource) and
ActiveDataSet.Edit;	(ADOSource.LockType =
Field.Clear;	ItBatchOptimistic);
end;	CancelBatch.Enabled :=
	BatchUpdate.Enabled;
procedure	ClearField.Enabled := Enabled;
TADODBTest.RefreshDataExecute(	RefreshData.Enabled := Enabled;
Sender: TObject);	ACTIONID attichmental : Diabicu,
<i>J / /</i>	

TBlobField(MasterGrid.SelectedField).LoadFromFile(OpenDialog.FileName); end; end; findexes }  procedure TADODBTest.IndexPageShow(Send	egin ActiveDataSet.Refresh; ad; cocedure ADODBTest.BinaryGetText(Sende TField; var Text: string; DisplayText: Boolean); egin Fext := Sender.AsString; ad;
d).LoadFromFile(OpenDialog.FileN ame); end; pr {Indexes}  procedure TADODBTest.IndexPageShow(Send er: TObject); end end; procedure TADODBTest.IndexPageShow(Send end) end	rocedure ADODBTest.BinaryGetText(Sende TField; var Text: string; DisplayText: Boolean); egin Fext := Sender.AsString; nd;
ame); end;  [Indexes]  procedure TADODBTest.IndexPageShow(Send er: TObject);  procedure end;  procedure TADODBTest.IndexPageShow(Send end)	rocedure ADODBTest.BinaryGetText(Sende TField; var Text: string; DisplayText: Boolean); egin Fext := Sender.AsString; nd;
end;  { Indexes }  procedure TADODBTest.IndexPageShow(Send er: TObject);  procedure Table PageShow(Send end er: Tobject);	ADODBTest.BinaryGetText(Sende TField; var Text: string; DisplayText: Boolean); egin Fext := Sender.AsString; nd;
{ Indexes }  procedure TADODBTest.IndexPageShow(Send er: TObject);  TADODBTEST.IndexPageShow(Send end end end end end end end end end	ADODBTest.BinaryGetText(Sende TField; var Text: string; DisplayText: Boolean); egin Fext := Sender.AsString; nd;
{ Indexes }  procedure TADODBTest.IndexPageShow(Send er: TObject);  r: be Table Tabl	TField; var Text: string; DisplayText: Boolean); egin Fext := Sender.AsString; nd;
procedure TADODBTest.IndexPageShow(Send er: TObject);  Description:	DisplayText: Boolean); egin Fext := Sender.AsString; nd;
procedure TADODBTest.IndexPageShow(Send er: TObject); be Table to	egin  Fext := Sender.AsString; ad;
TADODBTest.IndexPageShow(Send er: TObject);	Text := Sender.AsString; nd;
er: TObject);	ıd;
DESIII	,
	rocedure
, , , , , , , , , , , , , , , , , , , ,	ADODBTest.BinarySetText(Sende
	TField; const Text: string);
1 T - T.	egin
	Sender.AsString := Text;
	id;
CII	iu,
procedure	rocedure
F -	ADODBTest.BlobAsImageUpdate(
	ender: TObject);
	egin
9 /	BlobAsImage.Enabled :=
, ,	ssigned(ActiveDataSource.DataSet
	and
	ctiveDataSource.DataSet.Active
MasterClientData then an	
	(MasterGrid.SelectedField is
	BlobField);
	nd;
IndexDefs :=	
	rocedure
, , ,	ADODBTest.BlobAsImageExecute
	ender: TObject);
	egin
	BlobCtrlPanel.Visible := True;
	OBImage1.Visible := True;
` '	DBImage1.DataField :=
	lasterGrid.SelectedField.FieldNam
Exit; e;	
	nd;
for I := 0 to IndexDefs.Count - 1 do	,
	rocedure
	ADODBTest.LoadBlobFromFileEx
	eute(Sender: TObject);
` - ' /	egin
	OpenDialog.FilterIndex := 3;
	f OpenDialog.Execute then

end;	ame);
ciiu,	if IndexList.Items.Count > 0 then
procedure	begin
TADODBTest.IndexListClick(Sende	if (ADOSource = MasterTable)
r: TObject);	and
9 / /	(IndexList.Items.IndexOf(MasterTa
begin ShowIndexParams;	· · · · · · · · · · · · · · · · · · ·
if ActiveDataSet is TADOTable	ble.IndexName) > 0) then IndexList.ItemIndex :=
then	IndexList.Items.IndexOf(MasterTab
	le.IndexName) else
with TADOTable(ActiveDataSet)	IndexList.ItemIndex := 0;
do	ShowIndexParams;
begin	,
try	end;
{ Only Jet 4 supports setting	end;
indexname while open }	
MasterTable.IndexName :=	procedure TADODPTost ShowIndovPovoms
IndexList.Items[IndexList.ItemInde	TADODBTest.ShowIndexParams;
x];	var
except	IndexDef: TIndexDef;
Close;	begin
MasterTable.IndexName :=	if ActiveDataSource.DataSet is
IndexList.Items[IndexList.ItemInde	TADOTable then
x];	IndexDef :=
OpenTableExecute(nil);	TADOTable(ActiveDataSource.Data
end;	Set).IndexDefs[IndexList.ItemIndex]
end;	else
end;	if ActiveDataSource.DataSet is
_	TADODataSet then
procedure	IndexDef :=
TADODBTest.GridTitleClick(Colu	TADODataSet(ActiveDataSource.Da
mn: TColumn);	taSet).IndexDefs[IndexList.ItemInde
var	x
DataSet: TDataSet;	else
begin	Exit;
if not FMovingColumn then	idxCaseInsensitive.Checked :=
begin	ixCaseInsensitive in
DataSet := Column.Field.DataSet;	IndexDef.Options;
if DataSet is	idxDescending.Checked :=
TCustomADODataSet then	ixDescending in IndexDef.Options;
with	idxUnique.Checked := ixUnique in
TCustomADODataSet(DataSet) do	IndexDef.Options;
begin	idxPrimary.Checked := ixPrimary
if (Pos(Column.Field.FieldName,	in IndexDef.Options;
Sort) = 1) and (Pos(' DESC', Sort) =	IndexFields.Text :=
0) then	IndexDef.Fields;
Sort :=	DescFields.Text :=
Column.Field.FieldName + ' DESC'	IndexDef.DescFields;
else	CaseInsFields.Text :=
Sort :=	IndexDef.CaseInsFields;

Easure 24/10/2-0/20/20/101	Column Field FieldName   1 ACC!
Format('%s=%s%s%1:s',	Column.Field.FieldName + 'ASC';
[Field.FullName, QuoteChar,	StatusMsg := 'Sorted on '+Sort;
LocValue]);	end;
end;	end;
end;	FMovingColumn := False;
	end;
procedure	
TADODBTest.FilterKeyPress(Sende	procedure
r: TObject; var Key: Char);	TADODBTest.MasterGridColumn
begin	Moved(Sender: TObject;
FilterGroupBox.ItemIndex := -1;	FromIndex,
if Key = #13 then	ToIndex: Integer);
FilteredClick(Sender);	begin
end;	FMovingColumn := True;
,	end;
procedure	,
TADODBTest.FilterExit(Sender:	{ Filters }
TObject);	(210010)
begin	procedure
if Assigned(FActiveDataSet) then	TADODBTest.FilterPageShow(Send
ActiveDataSet.Filter :=	er: TObject);
Filter.Text;	
end;	var
ciiu,	Field: TField;
nuo o o duno	LocValue,
procedure TADODPTost Filtered Click Sendon	QuoteChar: string;
TADODBTest.FilteredClick(Sender:	begin
TObject);	if (Filter.Text = '') and
begin	Assigned(FActiveDataSet) and
if Filtered.Checked then	ActiveDataSet.Active then
ActiveDataSet.Filter :=	begin
Filter.Text;	Field :=
ActiveDataSet.Filtered :=	MasterGrid.SelectedField;
Filtered.Checked;	if Field = nil then Exit;
end;	with ActiveDataSet do
	try
procedure	DisableControls;
TADODBTest.FindFirstClick(Sende	MoveBy(3);
r: TObject);	LocValue :=
begin	VarToStr(Field.Value);
ActiveDataSet.Filter := Filter.Text;	First;
ActiveDataSet.FindFirst;	finally
end;	EnableControls;
	end;
procedure	if Field.DataType in [ftString,
TADODBTest.FindNextClick(Sende	ftMemo, ftFixedChar  then
r: TObject);	QuoteChar := "" else
begin	QuoteChar := ";
if ActiveDataSet.Filter <>	
Filter.Text then	Filter.Text :=
I HULL, I CAL UICH	

Field.FieldName;	ActiveDataSet.Filter :=
with ActiveDataSet do	Filter.Text;
try	ActiveDataSet.FindNext;
DisableControls;	end;
	cnu,
MoveBy(3); LocateEdit.Text :=	procedure
	procedure TADODBTest.FilterGroupBoxClick
VarToStr(Field.Value);	(Sender: TObject);
First;	begin
finally	if not Assigned(ADOSource) then
EnableControls;	Exit;
end;	case FilterGroupBox.ItemIndex of
end;	
end;	0: ADOSource.FilterGroup :=
_	fgPendingRecords;
procedure	1: ADOSource.FilterGroup :=
TADODBTest.LocateFieldDropDow	fgAffectedRecords;
n(Sender: TObject);	2: ADOSource.FilterGroup :=
begin	fgFetchedRecords;
	3: ADOSource.FilterGroup :=
ActiveDataSet.GetFieldNames(Locat	fgPendingRecords;
eField.Items);	else
end;	ADOSource.FilterGroup :=
	fgNone;
procedure	end;
TADODBTest.LocateButtonClick(Se	end;
nder: TObject);	
function LocateValue: Variant;	{ Locate }
var	
I: Integer;	procedure
Values: TStringList;	TADODBTest.LocatePageShow(Sen
begin	der: TObject);
if LocateNull.Checked then	var
Result := Null	Field: TField;
else if Pos(',', LocateEdit.Text) < 1	begin
then	if (FActiveDataSet <> nil) and
LocateValue := LocateEdit.Text	ActiveDataSet.Active then
else	begin
begin	Field :=
Values := TStringList.Create;	MasterGrid.SelectedField;
try	if LocateField.Items.Count = 0
Values.CommaText :=	then
LocateEdit.Text;	I ABILID DO A ABILID
Result :=	LocateFieldDropDown(LocateField)
VarArrayCreate([0,Values.Count-1],	;
varVariant);	if (LocateField.Text = ") or
for I := 0 to Values.Count - 1 do	(LocateField.Items.IndexOf(Field.Fi
Result[I] := Values[I];	eldName) < 1) then
	LocateField.Text :=

4 (C. 1. TOLL )	e n
xecute(Sender: TObject);	finally
begin	Values.Free;
StatusMsg := 'ApplyUpdates: '+	end;
IntToStr(MasterClientData.ApplyU	end;
pdates(-1));	end;
Beep;	
end;	var
	Options: TLocateOptions;
procedure	begin
TADODBTest.MidasCancelUpdates	Options := [];
Execute(Sender: TObject);	if locPartialKey.Checked then
begin	Include(Options, loPartialKey);
MasterClientData.CancelUpdates;	if
StatusMsg := 'Updates canceled';	ActiveDataSet.Locate(LocateField.T
end;	· ·
ciiu,	ext, LocateValue, Options) then
	StatusMsg := 'Record Found' else
procedure	StatusMsg := 'Not found';
TADODBTest.MasterClientDataRec	end;
oncileError(	
DataSet: TCustomClientDataSet;	procedure
E: EReconcileError; UpdateKind:	TADODBTest.LocateNullClick(Send
TUpdateKind;	er: TObject);
var Action: TReconcileAction);	begin
begin	LocateEdit.Enabled := not
Action :=	LocateNull.Checked;
HandleReconcileError(DataSet,	end;
UpdateKind, E);	
end;	{ Midas Testing }
	, , , , , , , , , , , , , , , , , , , ,
{ FieldSchema }	procedure
,	TADODBTest.ADOButtonClick(Sen
procedure	der: TObject);
TADODBTest.FieldsPageShow(Send	begin
er: TObject);	ActiveDataSet := ADOSource;
begin	,
CheckConnection(False);	end;
CheckConnection(Faise);	1
Connection Onen Scheme (ci Column	procedure
Connection.OpenSchema(siColumns	TADODBTest.MidasButtonClick(Se
, VarArrayOf([Unassigned,	nder: TObject);
Unassigned,	begin
MasterTableName.Text,	if Assigned(ADOSource) or
Unassigned]), EmptyParam,	MasterClientData.Active then
FieldSchema);	ActiveDataSet :=
end;	MasterClientData;
	end;
procedure	
TADODBTest.FieldSchemaDATA_	procedure
TYPEGetText(Sender: TField;	TADODBTest.MidasApplyUpdatesE
var Text: String; DisplayText:	111 022 1 continuos apply o puntos
8/ T J =	

'adVarBinary';
\$00000CD: Text :=
'adLongVarBinary';
\$00000088: Text := 'adChapter';
\$0000040: Text := 'adFileTime';
\$0000089: Text :=
'adDBFileTime';
\$0000008A: Text :=
'adPropVariant';
\$000008B: Text :=
'adVarNumeric';
else
Text := ' <unknown>';</unknown>
end;
, and the second
end;
,
{ Event Logging }
( - / - / - / - / - / - / - / - / - / -
procedure
TADODBTest.LogEvent(const
EventStr: string;
Component: TComponent = nil);
var
ItemCount: Integer;
begin
if (csDestroying in
ComponentState) or not
Events. Visible then Exit;
if (Component <> nil) and
(Component.Name $\Leftrightarrow$ '') then
(Component: vame > ) then
Events.Items.Add(Format('%s(%s)',
[EventStr, Component.Name])) else
Events.Items.Add(EventStr);
ItemCount := Events.Items.Count;
Events.ItemIndex := ItemCount - 1;
if ItemCount >
(Events.ClientHeight div
Events.ItemHeight) then  Events TenIndex = ItemCount
Events.TopIndex := ItemCount -
1;
end;
procedure
TADODBTest.ClearEventLogExecu
te(Sender: TObject);
begin

Boolean);
begin
case
FieldSchemaData_TYPE.Value of
<b>\$00000000:</b> Text := 'adEmpty';
<b>\$00000010:</b> Text := 'adTinyInt';
<b>\$00000002:</b> Text := 'adSmallInt';
<b>\$00000003:</b> Text := 'adInteger';
<b>\$00000014:</b> Text := 'adBigInt';
\$0000011: Text :=
'adUnsignedTinyInt';
<b>\$0000012:</b> Text :=
'adUnsignedSmallInt';
<b>\$00000013:</b> Text :=
'adUnsignedInt';
<b>\$00000015:</b> Text :=
'adUnsignedBigInt';
<b>\$00000004:</b> Text := 'adSingle';
<b>\$00000005:</b> Text := 'adDouble';
<b>\$00000006:</b> Text := 'adCurrency';
<b>\$0000000E:</b> Text := 'adDecimal';
<b>\$00000083:</b> Text := 'adNumeric';
\$000000B: Text := 'adBoolean';
\$000000A: Text := 'adError';
\$0000084: Text :=
'adUserDefined';
<b>\$000000C:</b> Text := 'adVariant';
<b>\$00000009:</b> Text := 'adIDispatch';
\$000000D: Text :=
'adIUnknown';
<b>\$00000048:</b> Text := 'adGUID';
\$0000007: Text := 'adDate';
\$00000085: Text := 'adDBDate';
\$00000086: Text := 'adDBTime';
\$0000087: Text :=
'adDBTimeStamp';
\$0000008: Text := 'adBSTR';
\$0000081: Text := 'adChar';
\$00000C8: Text := 'adVarChar';
\$00000C9: Text :=
'adLongVarChar';
\$00000082: Text := 'adWChar';
\$00000CA: Text :=
'adVarWChar';
\$000000CB: Text :=
'adLongVarWChar';
\$0000080: Text := 'adBinary';
\$00000CC: Text :=
ψυσυσυστι τελί:

Events.Items.Clear;   end;   end;		A VELLA LIETTA L'EST.
procedure TADODBTest.DataSetBeforeDelete( DataSet: TDataSet); begin LogEvent('BeforeDelete', DataSet); end;  procedure TADODBTest.DataSetBeforeEdit(D ataSet: TDataSet); begin LogEvent('BeforeEdit', DataSet); end;  procedure TADODBTest.DataSetBeforeEdit(D ataSet: TDataSet); begin LogEvent('BeforeEdit', DataSet); end;  procedure TADODBTest.DataSetBeforeInsert( DataSet: TDataSet); begin if Hook then begin DataSet.OnFetchComplete := DataSetFetchComplete; DataSetFetchProgress := DataSetFetchProgress; end else		
procedure TADODBTest.DataSetBeforeDelete( DataSet: TDataSet); begin  LogEvent('BeforeDelete', DataSet); end;  procedure TADODBTest.DataSetBeforeEdit(D ataSet: TDataSet); begin  LogEvent('BeforeEdit', DataSet); end;  procedure TADODBTest.DataSetBeforeEdit(D ataSet: TDataSet); begin  LogEvent('BeforeEdit', DataSet); end;  procedure TADODBTest.DataSetBeforeInsert( DataSet: TDataSet); begin  DataSet.OnFetchComplete := DataSetFetchComplete; DataSetFetchProgress := DataSetFetchProgress; end else	9 ,	enu;
TADODBTest.DataSetBeforeDelete( DataSet: TDataSet); begin  LogEvent('BeforeDelete', DataSet); end;  procedure TADODBTest.DataSetBeforeEdit(D ataSet: TDataSet); begin  LogEvent('BeforeEdit', DataSet); end;  procedure TADODBTest.DataSetBeforeEdit(D ataSet: TDataSet); begin  procedure TADODBTest.DataSetBeforeEdit(D ataSet: TDataSet); begin  procedure TADODBTest.DataSetBeforeInsert( DataSet: TDataSet); begin  DataSet.OnFetchComplete := DataSetFetchComplete;  DataSet.OnFetchProgress := DataSetFetchProgress; end else	na;	
TADODBTest.DataSetBeforeDelete( DataSet: TDataSet); begin  LogEvent('BeforeDelete', DataSet); end;  procedure  TADODBTest.DataSetBeforeEdit(D ataSet: TDataSet); begin  LogEvent('BeforeEdit', DataSet); end;  procedure  TADODBTest.DataSetBeforeInsert( DataSet: TDataSet); begin  procedure  TADODBTest.DataSetBeforeInsert( DataSet: TDataSet); begin  procedure  TADODBTest.DataSetBeforeInsert( DataSet: TDataSet); begin  DataSet.OnFetchComplete:  DataSet.OnFetchProgress:=  DataSetFetchProgress:=  DataSetFetchProgress; end else		
DataSet: TDataSet);   begin   ClearEventLog.Enabled :=   Events.Visible and (Events.Items.Count > 0);   end;     procedure   TADODBTest.DataSetBeforeEdit(D ataSet: TDataSet);   begin   TADODBTest.SetRecordSetEvents   Hook: Boolean; DataSet: TCustomADODataSet);   begin   if Hook then   begin   DataSet.OnFetchComplete :=   DataSetFetchComplete;   DataSet.OnFetchProgress :=   DataSetFetchProgress;   end;   else		
ClearEventLog.Enabled := Events.Visible and (Events.Items.Count > 0); end;	· •	•
Events.Visible and (Events.Items.Count > 0); end;	,,	
end; end;  procedure TADODBTest.DataSetBeforeEdit(D ataSet: TDataSet); begin LogEvent('BeforeEdit', DataSet); end;  procedure TADODBTest.DataSetBeforeInsert( DataSet: TDataSet); begin  TADODBTest.DataSetBeforeInsert( DataSet: TDataSet); begin LogEvent('BeforeInsert', DataSet); end;  LogEvent('BeforeInsert', DataSet); end;  [Events.Items.Count > 0); end;  procedure TADODBTest.SetRecordSetEvents Hook: Boolean; DataSet: TCustomADODataSet); begin if Hook then begin DataSet.OnFetchComplete := DataSetFetchComplete;  DataSetFetchProgress := DataSetFetchProgress; end else		
procedure TADODBTest.DataSetBeforeEdit(D ataSet: TDataSet); begin LogEvent('BeforeEdit', DataSet); end;  procedure TADODBTest.DataSetBeforeInsert( DataSet: TDataSet); begin  procedure TADODBTest.DataSetBeforeInsert( DataSet: TDataSet); begin  DataSet.OnFetchComplete := DataSetFetchComplete;  DataSetOnFetchProgress := DataSetFetchProgress; end;	LogEvent('BeforeDelete', DataSet);	_ ,
procedure TADODBTest.DataSetBeforeEdit(D ataSet: TDataSet); begin LogEvent('BeforeEdit', DataSet); end;  procedure TADODBTest.SetRecordSetEvents Hook: Boolean; DataSet: TCustomADODataSet); begin if Hook then begin DataSet.OnFetchComplete:= DataSetFetchComplete; DataSetFetchProgress:= DataSetFetchProgress; end; end else	nd;	(Events.Items.Count > 0);
TADODBTest.DataSetBeforeEdit(D ataSet: TDataSet); begin LogEvent('BeforeEdit', DataSet); end;  procedure TADODBTest.SetRecordSetEvents Hook: Boolean; DataSet: TCustomADODataSet); begin if Hook then begin DataSet.OnFetchComplete:= DataSetFetchComplete; DataSetOnFetchProgress:= DataSetFetchProgress:= DataSetFetchProgress; end;		end;
ataSet: TDataSet); begin  LogEvent('BeforeEdit', DataSet); end;  procedure TADODBTest.DataSetBeforeInsert( DataSet: TDataSet); begin  LogEvent('BeforeInsert', DataSet); end;  TADODBTest.SetRecordSetEvents Hook: Boolean; DataSet: TCustomADODataSet); begin if Hook then begin DataSet.OnFetchComplete:= DataSetFetchComplete; DataSetFetchProgress:= DataSetFetchProgress; end else	rocedure	
begin LogEvent('BeforeEdit', DataSet); end;  procedure TADODBTest.DataSetBeforeInsert( DataSet: TDataSet); begin  DataSet.OnFetchComplete:= DataSetFetchComplete; DataSetFetchProgress:= DataSetFetchProgress; end; end;	ADODBTest.DataSetBeforeEdit(D	procedure
LogEvent('BeforeEdit', DataSet); end;  procedure TADODBTest.DataSetBeforeInsert( DataSet: TDataSet); begin  DataSetConFetchComplete := DataSetFetchComplete; DataSetFetchProgress := DataSetFetchProgress; end; end else		TADODBTest.SetRecordSetEvents(
LogEvent('BeforeEdit', DataSet); end;  procedure TADODBTest.DataSetBeforeInsert( DataSet: TDataSet); begin  DataSetConFetchComplete := DataSetFetchComplete;  DataSetOnFetchProgress := DataSetFetchProgress; end; end else	egin	Hook: Boolean; DataSet:
end;  begin  if Hook then  begin  TADODBTest.DataSetBeforeInsert( DataSet: TDataSet);  begin  DataSet.OnFetchComplete := DataSetFetchComplete;  DataSet.OnFetchProgress := DataSetFetchProgress; end;  end else		TCustomADODataSet);
if Hook then  begin  DataSet: TDataSet);  begin  LogEvent('BeforeInsert', DataSet);  end;  if Hook then  begin  DataSet.OnFetchComplete :=  DataSetFetchComplete;  DataSetFetchProgress :=  DataSetFetchProgress;  end  else		begin
TADODBTest.DataSetBeforeInsert( DataSet: TDataSet);  begin  LogEvent('BeforeInsert', DataSet); end;  DataSet.OnFetchComplete:  DataSetFetchComplete:  DataSetFetchProgress:  end else	,	if Hook then
TADODBTest.DataSetBeforeInsert( DataSet: TDataSet);  begin  LogEvent('BeforeInsert', DataSet); end;  DataSet.OnFetchComplete:  DataSetFetchComplete:  DataSetFetchProgress:  end else	rocedure	begin
DataSet: TDataSet); begin LogEvent('BeforeInsert', DataSet); end; DataSetFetchComplete; DataSetFetchProgress := DataSetFetchProgress; end else		
begin LogEvent('BeforeInsert', DataSet); end; DataSet.OnFetchProgress := DataSetFetchProgress; end else	`	<u> </u>
LogEvent('BeforeInsert', DataSet); end; DataSetFetchProgress; end else	,,	
end; end else	S .	S
else	•	
	iru,	
procedure     Degin	rocedure	
r		DataSet.OnFetchComplete := nil;
,	· ·	DataSet.OnFetchProgress := nil;
	, .	
· · · · · · · · · · · · · · · · · · ·	ĕ	
LogEvent('BeforePost', DataSet); end;		ciiu,
end;	nu;	nucacduuc
procedure  TA DODDT act Data Sat Bafava Class		<del>-</del> -
		TADODBTest.DataSetBeforeClose(
TADODBTest.DataSetBeforeScroll( DataSet: TDataSet);	· · · · · · · · · · · · · · · · · · ·	
DataSet: TDataSet); begin	, ,	
begin LogEvent('BeforeClose');	3	
LogEvent('BeforeScroll', DataSet); end;		end;
end;	nd;	
procedure		<del>-</del> -
•		TADODBTest.DataSetAfterScroll(D
TADODBTest.DataSetCalcFields(Da ataSet: TDataSet);	· ·	
taSet: TDataSet); begin	Set: TDataSet);	8
		LogEvent('AfterScroll', DataSet);
LogEvent('OnCalcFields', end;	LogEvent('OnCalcFields',	end;
DataSet);		
end; procedure		procedure
		TADODBTest.DataSetBeforeCancel
procedure (DataSet: TDataSet);	rocedure	

	TADODRE AD A CAE AD A C
begin	TADODBTest.DataSetError(DataSe
LogEvent('AfterCancel', DataSet);	t: TDataSet;
end;	E: EDatabaseError; var Action:
	TDataAction);
procedure	begin
TADODBTest.DataSourceStateCha	
nge(Sender: TObject);	LogEvent('OnDelete/OnEdit/OnPost
begin	Errors', DataSet);
LogEvent('OnStateChange',	end;
Sender as TComponent);	
end;	procedure
,	TADODBTest.DataSetNewRecord(D
procedure	ataSet: TDataSet);
TADODBTest.DataSourceUpdateDa	begin
ta(Sender: TObject);	LogEvent('OnNewRecord',
begin	DataSet);
LogEvent('OnUpdateData', Sender	end;
as TComponent);	chu,
end;	procedure
CHU,	TADODBTest.DataSetAfterPost(Dat
	aSet: TDataSet);
procedure	, , , , , , , , , , , , , , , , , , ,
TADODBTest.MasterTableBeforeSc	begin
roll(DataSet: TDataSet);	LogEvent('AfterPost', DataSet);
begin	end;
LogEvent('BeforeScroll', DataSet);	
end;	procedure
	TADODBTest.DataSetAfterInsert(D
procedure	ataSet: TDataSet);
TADODBTest.MasterTableAfterScr	begin
oll(DataSet: TDataSet);	LogEvent('AfterInsert', DataSet);
begin	end;
LogEvent('AfterScroll', DataSet);	
end;	procedure
	TADODBTest.DataSetAfterEdit(Dat
procedure	aSet: TDataSet);
TADODBTest.OnFilterRecord(Data	begin
Set: TDataSet; var Accept:	LogEvent('AfterEdit', DataSet);
Boolean);	end;
begin	
Accept :=	procedure
(DataSet.Fields[0].AsInteger = 2);	TADODBTest.DataSetAfterDelete(D
end;	ataSet: TDataSet);
	begin
procedure	LogEvent('AfterDelete', DataSet);
TADODBTest.DataSetFetchProgres	end;
s(DataSet: TCustomADODataSet;	cnu,
	nuocoduno
Progress, MaxProgress: Integer;	procedure TADODPTost DataSatAftarCanasl(
var EventStatus: TEventStatus);	TADODBTest.DataSetAfterCancel(
begin	DataSet: TDataSet);

C	IE4(E-4-1-D
Connection);	LogEvent(Format('FetchProgress:
end;	%d of %d', [Progress,
	MaxProgress]), DataSet);
procedure	end;
TADODBTest.ConnectionDisconnec	
t(Connection: TADOConnection;	procedure
var EventStatus: TEventStatus);	TADODBTest.FieldValidate(Sender:
begin	TField);
LogEvent('Disconnect',	begin
Connection);	LogEvent(Format('Val: %s=%s',
end;	[Sender.DisplayName,
	Sender.AsSTring]));
procedure	end;
TADODBTest.ConnectionExecuteC	
omplete(Connection:	{ Connection Events }
TADOConnection;	,
RecordsAffected: Integer; const	procedure
Error: Error;	TADODBTest.ConnectionBeginTra
var EventStatus: TEventStatus;	nsComplete(
const Command: Command;	Connection: TADOConnection;
const Recordset: Recordset);	TransactionLevel: Integer;
begin	const Error: Error; var
LogEvent('ExecuteComplete',	EventStatus: TEventStatus);
§ ` ` •	begin
Connection);	Č
end;	LogEvent('BeginTransComplete',
	Connection);
procedure	end;
TADODBTest.ConnectionInfoMessa	1
ge(Connection: TADOConnection;	procedure
const Error; Error; var	TADODBTest.ConnectionCommitT
EventStatus: TEventStatus);	ransComplete(Connection:
begin	TADOConnection;
LogEvent('InfoMessage',	const Error: Error; var
Connection);	EventStatus: TEventStatus);
end;	begin
procedure	LogEvent('CommitTransComplete',
TADODBTest.ConnectionRollbackT	Connection);
ransComplete(	end;
Connection: TADOConnection;	
const Error: Error;	procedure
var EventStatus: TEventStatus);	TADODBTest.ConnectionConnectC
begin	omplete(Connection:
	TADOConnection;
LogEvent('RollbackTransComplete',	const Error: Error; var
Connection);	EventStatus: TEventStatus);
end;	begin
,	ClearProgressBar;
procedure	LogEvent('ConnectComplete',
procedure	· · · · · · · · · · · · · · · · · ·

MasterQuery.Parameters else	TADODBTest.ConnectionWillConn
FParamSource :=	ect(Connection: TADOConnection;
MasterProc.Parameters;	var ConnectionString, UserID,
,	Password: WideString;
if not Showing then Exit; UpdateParameterList;	var ConnectOptions:
	TConnectOption; var EventStatus:
end;	TEventStatus);
1	begin
procedure	
TADODBTest.RefreshParameters(P	LogEvent('WillConnect', Connection);
arameters: TParameters);	
Var	end;
I: Integer;	nuocoduus
NewParameter: TParameter;	procedure TADODBTest.ConnectionWillExecu
begin	
try	te(Connection: TADOConnection;
Parameters.Refresh;	var CommandText: WideString;
except	var CursorType: TCursorType;
end;	var LockType: TADOLockType;
if Parameters.Count = 0 then Exit;	var CommandType:
for I := 0 to Parameters.Count - 1	TCommandType;
do	var ExecuteOptions:
with Parameters[I] do	TExecuteOptions; var EventStatus:
if Name[1] = '@' then	TEventStatus;
begin	const Command: _Command;
NewParameter :=	const Recordset: _Recordset);
Parameters.CreateParameter(Copy(	begin
Name, 2, 100), DataType, Direction,	LogEvent('WillExecute',
Size, Null);	Connection);
NewParameter.Index := I;	end;
Parameters[I].Free;	
end;	procedure
end;	TADODBTest.ConnectionLogin(Sen
,	der: TObject; Username,
procedure	Password: String);
TADODBTest.ParamPageShow(Sen	begin
der: TObject);	LogEvent(Format('OnLogin - UID:
var	%s PWD: %s',[UserName,
FT: TFieldType;	Password]), Sender as
begin	TADOConnection);
if ParameterType.Items.Count = 0	end;
then	
with ParameterType.Items do	{ Parameters }
for FT := low(TFieldType) to	
high(TFieldType) do	procedure
Add(FieldTypeNames[FT]);	TADODBTest.ParameterSourceClic
end;	k(Sender: TObject);
ciiu,	begin
procedure	if SQLParams.Checked then
procedure	FParamSource :=

procedure	TADODBTest.UpdateParameterList
TADODBTest.AddParameterButton	;
Click(Sender: TObject);	var
begin	I: Integer;
	begin
FParamSource.CreateParameter('P	with ParameterList.Items do
aram'+IntToStr(FParamSource.Cou	try
nt+1), ftInteger, pdInput, 0, 0);	BeginUpdate;
FModifiedParameter := -1;	Clear;
UpdateParameterList;	for I := 0 to FParamSource.Count
end;	- 1 do
procedure	Add(FParamSource[I].DisplayName
TADODBTest.ParameterListClick(S	);
ender: TObject);	if ParameterList.Items.Count > 0
begin	then
WriteParameterData;	begin
if ParameterList.ItemIndex < 0	if FModifiedParameter > -1 then
then Exit;	ParameterList.ItemIndex :=
with	FModifiedParameter else
FParamSource[ParameterList.ItemI	ParameterList.ItemIndex := 0;
ndex] do	,
begin	ParameterListClick(ParameterList);
ParameterName.Text := Name;	end else
ParameterValue.Text :=	begin
VarToStr(Value);	ParameterName.Text := ";
ParameterType.Text :=	ParameterValue.Text := ";
FieldTypeNames[DataType];	end;
ParameterSize.Text :=	finally
IntToStr(Size);	EndUpdate;
ParameterScale.Text :=	end;
IntToStr(NumericScale);	end;
ParameterPrecision.Text :=	chu,
IntToStr(Precision);	procedure
( 11 11 1)	TADODBTest.RefreshParametersB
ParameterDirectionGroup.ItemInde	uttonClick(Sender: TObject);
x := Ord(Direction)-1;	begin
PANullableCheckbox.Checked :=	CheckConnection(False);
paNullable in Attributes;	if SQLParams.Checked then
PALongCheckbox.Checked :=	
paLong in Attributes;	MasterQuery.SQL.Text :=
PASignedCheckbox.Checked :=	MasterSQL.Text else
paSigned in Attributes;	MasterProc.ProcedureName :=
end;	MasterProcName.Text;
FModifiedParameter := -1;	Defuel Deve to(ED
end;	RefreshParameters(FParamSource);
	UpdateParameterList;
procedure	end;
TADODBTest.WriteParameterData;	
TADODDICSI. WITHER AT AIRCICI DATA,	

	var
procedure	DataTypeIndex: Integer;
TADODBTest.ParameterDataChang	begin
e(Sender: TObject);	if FModifiedParameter < 0 then
begin	Exit;
FModifiedParameter :=	with
ParameterList.ItemIndex;	
	FParamSource[FModifiedParameter   1 do
end;	begin
	if Name <> ParameterName.Text
procedure	
TADODBTest.MasterSQLChange(S	then
ender: TObject);	begin
begin	Name := ParameterName.Text;
ParameterList.Items.Clear;	D A TO A TABLE OF THE STATE OF
end;	ParameterList.Items[FModifiedPara
	meter] := Name;
{ Test Code }	end;
	DataTypeIndex :=
procedure	ParameterType.Items.IndexOf(Para
TADODBTest.TestButtonClick(Sen	meterType.Text);
der: TObject);	if DataTypeIndex <> -1 then
begin	DataType :=
{ Put your test code here }	TFieldtype(DataTypeIndex) else
end;	DataType := ftInteger;
	Size :=
	StrToInt(ParameterSize.Text);
procedure	NumericScale :=
TADODBTest.ToolBar1Click(Sende	StrToInt(ParameterScale.Text);
r: TObject);	Precision :=
begin	StrToInt(ParameterPrecision.Text);
	Direction :=
end;	TParameterDirection(ParameterDir
	ectionGroup.ItemIndex+1);
end.	if PANullableCheckbox.Checked
	then
\\\\\\\\\	Attributes := [paNullable];
	if PALongCheckbox.Checked then
الملف النصي للبرنامج	Attributes := Attributes +
object ADODBTest: TADODBTest	[paLong];
Left = 175	if PASignedCheckbox.Checked
Top = 114	then
HorzScrollBar.Increment = 52	Attributes := Attributes +
VertScrollBar.Increment = 45	[paSigned];
AutoScroll = False	if VarToStr(Value) ↔
Caption = 'ADO DB Controls Test	ParameterValue.Text then
Application'	Value := ParameterValue.Text;
ClientHeight = 460	end;
ClientWidth = 680	FModifiedParameter := -1;
	end;

Anchors = [akLeft, akTop,
akRight]
Constraints.MinWidth = 500
TabOrder = 1
object SourcePage: TTabSheet
Caption = 'Source'
object EditConnStr:
TSpeedButton
Left = 460
<b>Top</b> = 14
Width = 18
Height = 18
Caption = ''
Font.Charset =
ANSI CHARSET
Font.Color = clWindowText
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
OnClick = EditConnStrClick
end
object Label2: TLabel
Left = 6
Top = -1
<b>Width = 87</b>
Height = 13
Caption = 'Connection String:'
Font.Charset =
ANSI CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
object TableNames: TGroupBox Left = 5
Top = 38
Width = 133
Height = 76
Caption = ' Master/Detail
Tables '
TabOrder = 1
object MasterTableName:
TComboBox
Left = 8
Top = 18

C. I. ID. E
Color = clBtnFace
ParentFont = True
Menu = MainMenu1
OldCreateOrder = True
Position = poScreenCenter
OnCloseQuery = FormCloseQuery
OnCreate = FormCreate
OnDestroy = FormDestroy
DesignSize = (
680
460)
PixelsPerInch = 96
TextHeight = 13
object Label1: TLabel
Left = 16
Top = 53
Width = 3
Height = 13
end
object StatusBar: TStatusBar
Left = 0
Top = 436
Width = 680
Height = 24
Panels = <
item
Text = 'Message'
Width = 300
end
0.22.02
item
Text = 'Modified'
Width = 80
end
item
Text = 'Null'
Width = 50
end
item
Text = 'RecNo'
Width = 50
end>
end
object AreaSelector: TPageControl
Left = 5
Top = 30
Width = 501
Height = 148
ActivePage = TabSheet1

Action = PrevQuery
Flat = True
Glyph.Data = {
ParentShowHint = False
ShowHint = True
end
object NextQuery1:
TSpeedButton
Left = 189
Top = 52
Width = 19
Height = 13
Action = NextQuery
Flat = True
Glyph.Data = {
ParentShowHint = False
ShowHint = True
end
object EditCommandText:
TSpeedButton
Left = 189
Top = 19
Width = 18
Height = 18
Caption = ''
Font.Charset =
ANSI_CHARSET
Font.Color = clWindowText
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
OnClick =
<b>EditCommandTextClick</b>
end
object MasterSQL: TMemo
Left = 7
Top = 18
Width = 180
Height = 21
Lines.Strings = (
'Select * from Customer')
TabOrder = 0
OnChange =
MasterSQLChange
OnKeyPress =
MasterSQLKeyPress
end

Width = 116
Height = 21
DropDownCount = 20
ItemHeight = 13
Sorted = True
TabOrder = 0
Text = 'Customer'
OnClick =
Master Table Name Click
OnDropDown =
MasterTableNameDropDown
OnKeyPress =
MasterTableNameKeyPress
end
object DetailTableName:
TComboBox
Left = 8
Top = 45
Width = 116
Height = 21
DropDownCount = 20
ItemHeight = 13
Sorted = True
TabOrder = 1
Text = 'Orders'
OnClick =
<b>DetailTableNameClick</b>
OnDropDown =
MasterTableNameDropDown
OnKeyPress =
MasterTableNameKeyPress
end
end
object QueryStrings:
TGroupBox
Left = 143
Top = 38
Width = 212
Height = 75
Caption = ' Master/Detail
Queries '
TabOrder = 2
object PrevQuery1:
TSpeedButton
Left = 189
Top = 39
Width = 19
Height = 13

Markan Dana Nama Citala
MasterProcNameClick
OnDropDown =
ProcNameDropDown
OnKeyPress =
MasterProcNameKeyPress
end
object DetailProcName:
TComboBox
Left = 9
Top = 44
Width = 108
Height = 21
ItemHeight = 13
TabOrder = 1
OnClick =
<b>DetailProcNameClick</b>
OnDropDown =
ProcNameDropDown
end
end
end
object FieldsPage: TTabSheet
Caption = 'Fields'
OnShow = FieldsPageShow
object FieldSchemaGrid:
TDBGrid
Left = 0
Top = 0
Width = 493
Height = 120
Align = alClient
DataSource =
FieldSchemaSource
TabOrder = 0
TitleFont.Charset =
DEFAULT CHARSET
TitleFont.Color =
clWindowText
TitleFont.Height = -11
TitleFont.Name = 'MS Sans
Serif'
TitleFont.Style = []
end
end
object ParamPage: TTabSheet
Caption = 'Parameters'
ImageIndex = 6
OnShow = ParamPageShow

object DetailSQL: TMemo
Left = 7
Top = 45
Width = 180
Height = 21
Lines.Strings = (
'select * from orders where
CustNo = '
':CustNo')
TabOrder = 1
OnKeyPress =
The state of the s
MasterSQLKeyPress end
end
object ConnectionString:
TComboBox
Left = 6
Top = 13
Width = 450
Height = 21
Hint = 'asdfasdf'
DropDownCount = 20
ItemHeight = 13
ParentShowHint = False
ShowHint = True
TabOrder = 0
OnClick =
ConnectionStringClick
OnKeyPress =
ConnectionStringKeyPress
end
object ProcedureNames:
TGroupBox
Left = 361
Top = 38
Width = 125
Height = 75
Caption = 'Stored Procedure '
TabOrder = 3
object MasterProcName:
TComboBox
Left = 9
<b>Top</b> = 17
Width = 108
Height = 21
ItemHeight = 13
TabOrder = 0
OnClick =
Onchek –

ItemHeight = 13
TabOrder = 0
OnClick = ParameterListClick
end
object ParameterName: TEdit
Left = 135
<b>Top = 16</b>
<b>Width = 101</b>
Height = 21
TabOrder = 1
OnChange =
Parameter Data Change
end
object ParameterValue: TEdit
Left = 135
Top = 93
Width = 101
Height = 21
TabOrder = 2
OnChange =
Parameter Data Change
end
object ParameterSize: TEdit
Left = 242
Top = 16
Width = 71
Height = 21
TabOrder = 3
OnChange =
Parameter Data Change
end
object ParameterScale: TEdit
Left = 242
Top = 55
Width = 71
Height = 21
TabOrder = 4
OnChange =
Parameter Data Change
end
object ParameterPrecision:
TEdit
Left = 242
Top = 93
<b>Width</b> = 71
Height = 21
TabOrder = 5
OnChange =

object ParameterNameLabel:
TLabel
Left = 135
Top = 2
<b>Width = 28</b>
Height = 13
Caption = 'Name'
end
object PTypeLabel: TLabel
Left = 135
Top = 40
Width = 24
Height = 13
Caption = 'Type'
end
object PValueLabel: TLabel
Left = 135
Top = 79
<b>Width = 27</b>
Height = 13
Caption = 'Value'
end
object PSizeLabel: TLabel
Left = 243
Top = 2
Width = 20
Height = 13
Caption = 'Size'
end
object PScaleLabel: TLabel
Left = 242
Top = 41
Width = 27
Height = 13
Caption = 'Scale'
end
object PPrecisionLabel: TLabel
Left = 242
Top = 79
Width = 43
Height = 13
Caption = 'Precision'
end
object ParameterList: TListBox
Left = 2
Top = 16
-
Width = 124
Height = 74

Parameter Pata Chango
ParameterDataChange
end
object PALongCheckBox:
TCheckBox
Left = 87
Top = 13
Width = 52
Height = 17
Caption = 'Long'
TabOrder = 2
OnClick =
Parameter Data Change
end
end
object AddParameterButton:
TButton
Left = 7
Top = 94
<b>Width = 54</b>
Height = 23
Caption = 'Add'
TabOrder = 8
OnClick =
AddParameterButtonClick
end
object
RefreshParametersButton: TButton
Left = 66
Top = 94
Width = 54
Height = 23
Caption = 'Refresh'
TabOrder = 9
OnClick =
RefreshParametersButtonClick
end
object ParameterType:
TComboBox
Left = 135
Top = 55
Width = 101
Height = 21
ItemHeight = 13
TabOrder = 10
OnChange =
ParameterDataChange
end
object SQLParams:

D C
Parameter Data Change
end
object
ParameterDirectionGroup:
TRadioGroup
Left = 322
Top = 2
Width = 162
Height = 57
Caption = ' Direction '
Columns = 2
Items.Strings = (
'Input'
1
'Output' 'In/Out'
'Return')
TabOrder = 6
OnClick =
<b>ParameterDataChange</b>
end
object ParamAttributes:
TGroupBox
Left = 322
Top = 63
Width = 162
Height = 52
Caption = ' Attributes '
TabOrder = 7
object PANullableCheckBox:
TCheckBox
Left = 11
Top = 13
Width = 72
Height = 17
Caption = 'Nullable'
TabOrder = 0
OnClick =
<b>ParameterDataChange</b>
end
object PASignedCheckBox:
TCheckBox
Left = 11
Top = 31
Width = 66
Height = 17
Caption = 'Signed'
TabOrder = 1
OnClick =
OHOHON

object Label13: TLabel
Left = 156
Top = 80
Width = 68
Height = 13
Caption = 'CaseInsFields:'
end
object IndexList: TListBox
Left = 8
Top = 17
Width = 137
Height = 100
ItemHeight = 13
TabOrder = 0
OnClick = IndexListClick
end
object IndexFields: TEdit
Left = 156
Top = 17
Width = 164
Height = 21
ReadOnly = True
TabOrder = 1
end
object DescFields: TEdit
Left = 156
Top = 56
Width = 164
Height = 21
ReadOnly = True
TabOrder = 2
end
object CaseInsFields: TEdit
Left = 156
Top = 96
Width = 164
Height = 21
ReadOnly = True
TabOrder = 3
end
object IndexOptions:
TGroupBox
Left = 333
Top = 2 Width = 159
Height = 113
Caption = 'Index Options'
TabOrder = 4

TD - 1: - D44
TRadioButton
Left = 5
Top = 1
Width = 43
Height = 14
Caption = 'SQL'
Checked = True
TabOrder = 11
TabStop = True
OnClick =
<b>ParameterSourceClick</b>
end
object ProcParams:
TRadioButton
Left = 50
<b>Top = 1</b>
Width = 73
Height = 14
Caption = 'StoredProc'
TabOrder = 12
OnClick =
ParameterSourceClick
end
end
object IndexPage: TTabSheet
Caption = 'Indexes'
OnShow = IndexPageShow
object Label11: TLabel
Left = 10
Top = 2
Width = 65
Height = 13
Caption = 'Index Names:'
end
object Label14: TLabel
Left = 156
Top = 3
10p = 3   Width = 30
Height = 13
Caption = 'Fields:'
end
object Label3: TLabel
Left = 156
Top = 40
Width = 55
Height = 13
Caption = 'DescFields:'
end

G
Caption = 'Filter:'
end
object Filter: TEdit
Left = 14
<b>Top = 17</b>
Width = 235
Height = 21
TabOrder = 0
OnExit = FilterExit
OnKeyPress = FilterKeyPress
end
object FindFirst: TButton
Left = 174
Top = 50
Width = 75
Height = 25
Caption = 'F&ind First'
TabOrder = 2
OnClick = FindFirstClick
end
object FindNext: TButton
Left = 174
Top = 84
Width = 75
Height = 25
Caption = 'Find &Next'
TabOrder = 3
OnClick = FindNextClick
end
object Filtered: TCheckBox
Left = 28
Top = 46
Width = 97
Height = 17
Caption = 'Filtered'
TabOrder = 1
OnClick = FilteredClick
end
object FilterGroupBox:
TRadioGroup
Left = 288
Top = 9
Width = 110
Height = 105
Caption = ' Filter Group '
Items.Strings = (
'Pending'
'Affected'

object idxCaseInsensitive:
TCheckBox
Left = 12
Top = 85
Width = 109
Height = 17
Caption = 'Case Insensitive'
Enabled = False
TabOrder = 0
end
object idxDescending:
TCheckBox
Left = 12
Top = 64
Width = 86
Height = 17
Caption = 'Descending'
Enabled = False
TabOrder = 1
end
object idxPrimary: TCheckBox
Left = 12
Top = 21
Width = 97
Height = 17
Caption = 'Primary'
Enabled = False
TabOrder = 2
end
object idxUnique: TCheckBox
Left = 12
Top = 42
<b>Width = 97</b>
Height = 17
Caption = 'Unique'
Enabled = False
TabOrder = 3
end
end
end
object FilterPage: TTabSheet
Caption = 'Filters'
OnShow = FilterPageShow
object Label10: TLabel
Left = 14
Top = 2
Width = 25
Height = 13

end
object LocateField:
TComboBox
Left = 10
Top = 32
Width = 183
Height = 21
ItemHeight = 13
Sorted = True
TabOrder = 0
OnDropDown =
LocateFieldDropDown
end
object locPartialKey:
TCheckBox
Left = 217
Top = 74
Width = 97
7.1 - 0.2 0.2
Height = 16
Caption = 'Partial Key' TabOrder = 4
end
object LocateNull: TCheckBox Left = 217
Top = 48 Width = 97
Height = 17
Caption = 'Null Value' TabOrder = 3
OnClick = LocateNullClick
end
end
end
object TabSheet1: TTabSheet
Caption = 'MiscTest'
object TestButton: TButton
Left = 8
Top = 7
Width = 75
Height = 25
Caption = 'Test'
TabOrder = 0
OnClick = TestButtonClick
end
end
end
object DataPanel: TPanel
Left = 5

'Fetched'
'Conflicting')
TabOrder = 4
OnClick =
FilterGroupBoxClick
end
end
object LocatePage: TTabSheet
Caption = 'Locate'
OnShow = LocatePageShow
object LocateGroupBox:
TGroupBox
Left = 4
Top = 1
Width = 329
Height = 114
Caption = ' Locate '
TabOrder = 0
object Label16: TLabel
Left = 10
Top = 16
Width = 36
Height = 13
Caption = 'Field(s):'
end
object Label17: TLabel
Left = 11
Top = 61
Width = 122
Height = 13
Caption = 'Values (comma
delimited):'
end
object LocateButton: TButton
Left = 216
Top = 18
Width = 75
Height = 25
Caption = '&Locate'
TabOrder = 2
OnClick = LocateButtonClick
end
object LocateEdit: TEdit
Left = 10
Top = 79
Width = 183
Height = 21
TabOrder = 1

Checked = True
TabOrder = 1
TabStop = True
OnClick = ADOButtonClick
end
object MidasButton:
TRadioButton
Left = 312
Top = 8
Width = 54
Height = 17
Caption = 'Midas'
TabOrder = 2
OnClick = MidasButtonClick
end
end
object BlobCtrlPanel: TPanel
Left = 315
Top = 30
Width = 185
Height = 219
Align = alRight
BevelOuter = bvNone
BorderWidth = 2
TabOrder = 2
Visible = False
object Splitter1: TSplitter
Left = 2
Top = 107 Width = 181
Height = 2
Cursor = crVSplit
Align = alBottom
Visible = False
end
object DBMemo1: TDBMemo
Left = 2
Top = 2
Width = 181
Height = 105
Align = alClient
DataSource =
MasterDataSource
MaxLength = 1024
PopupMenu = PopupMenu1
TabOrder = 0
end
object DBImage1: TDBImage

Top = 183
Width = 501
Height = 250
Anchors = [akLeft, akTop,
akRight, akBottom]
Constraints.MinHeight = 80
Constraints.MinWidth = 500
TabOrder = 2
object NavigatorPanel: TPanel
Left = 1
Top = 1
Width = 499
Height = 29
Align = alTop
BevelOuter = bvNone
TabOrder = 0
object ReadOnlyLabel: TLabel
Left = 403
Top = 9
Width = 60
Height = 13
Caption = 'Read Only'
Font.Charset =
ANSI CHARSET
Font.Color = clRed
Font.Height = -11 Font.Name = 'MS Sans Serif'
Font.Style = [fsBold] ParentFont = False
Visible = False
end
object DBNavigator1:
TDBNavigator
Left = 4
Top = 3
Width = 240
Height = 25
DataSource =
MasterDataSource
TabOrder = 0
end
object ADOButton:
TRadioButton
Left = 257
Top = 8
Width = 42
Height = 17
Caption = 'ADO'

OnTitleClick = GridTitleClick
end
object MasterGrid: TDBGrid
Left = 2
Top = 2
Width = 310
Height = 105
Align = alClient
DataSource =
MasterDataSource
ParentShowHint = False
ShowHint = True
TabOrder = 1
TitleFont.Charset =
DEFAULT CHARSET
TitleFont.Color =
clWindowText
TitleFont.Height = -11
TitleFont.Name = 'MS Sans
Serif'
TitleFont.Style = []
OnColEnter = GridColEnter
OnColumnMoved =
MasterGridColumnMoved
OnEnter = GridSetFocus
OnTitleClick = GridTitleClick
end
object DBEditScroller:
TScrollBox
Left = 2
Top = 2
Width = 303
Height = 0
TabOrder = 2
end
end
end
object Events: TListBox
Left = 512
Top = 50
Width = 164
Height = 381
Anchors = [akTop, akRight,
akBottom]
ItemHeight = 13
TabOrder = 3
end
object ToolBar1: TToolBar
V

Left = 2
Top = 109
Width = 181
Height = 108
Align = alBottom
DataSource =
Master Data Source
PopupMenu = PopupMenu1
TabOrder = 1
end
end
object GridPanel: TPanel
Left = 1
Top = 30
Width = 314
Height = 219
Align = alClient
BevelOuter = bvNone
BorderWidth = 2
TabOrder = 1
object GridSplitter: TSplitter
Left = 2
Top = 107
Width = 310
Height = 2
Cursor = crVSplit
Align = alBottom
Visible = False
end
object DetailGrid: TDBGrid
Left = 2
Top = 109
Width = 310
Height = 108
Align = alBottom
DataSource = DetailDataSource
TabOrder = 0
TitleFont.Charset =
DEFAULT_CHARSET
TitleFont.Color =
clWindowText
TitleFont.Height = -11
TitleFont.Name = 'MS Sans
Serif'
TitleFont.Style = []
Visible = False
OnColEnter = GridColEnter
OnEnter = GridSetFocus

end
object SavePacketButton: TToolButton
Left = 129
Top = 0
Action = SaveToFile
end
object Sep2: TToolButton
Left = 152
Top = 0
Width = 14
Caption = 'Sep2'
ImageIndex = 1
Style = tbsDivider
end
object ClearFieldButton:
TToolButton
Left = 166
Top = 0
Action = ClearField
end
object Sep3: TToolButton
Left = 189
Top = 0
Width = 14
Caption = 'Sep3'
ImageIndex = 1
Style = tbsDivider
end
object BatchUpdateButton:
TToolButton
Left = 203
Top = 0
Action = BatchUpdate
end
object CancelBatchButton:
TToolButton
Left = 226
Top = 0
Action = CancelBatch
end
object Sep4: TToolButton
Left = 249
Top = 0
Width = 14
Caption = 'Sep4'
ImageIndex = 21
Style = tbsSeparator
•

Left = 0
Top = 0
Width = 680
Height = 29
BorderWidth = 1
Caption = 'ToolBar1'
Flat = True
Images = ImageList1
ParentShowHint = False
ShowHint = True
TabOrder = 4
OnClick = ToolBar1Click
object OpenTableButton:
TToolButton
Left = 0
Top = 0
Action = OpenTable
end
object OpenQueryButton:
TToolButton
Left = 23
Top = 0
Action = OpenQuery
end
object ToolButton1: TToolButton
Left = 46
Left = 46 Top = 0
Left = 46 Top = 0 Action = OpenProcedure
Left = 46 Top = 0 Action = OpenProcedure end
Left = 46 Top = 0 Action = OpenProcedure end object CloseActiveButton:
Left = 46 Top = 0 Action = OpenProcedure end object CloseActiveButton: TToolButton
Left = 46 Top = 0 Action = OpenProcedure end object CloseActiveButton: TToolButton Left = 69
Left = 46 Top = 0 Action = OpenProcedure end object CloseActiveButton: TToolButton Left = 69 Top = 0
Left = 46 Top = 0 Action = OpenProcedure end object CloseActiveButton: TToolButton Left = 69
Left = 46 Top = 0 Action = OpenProcedure end object CloseActiveButton: TToolButton Left = 69 Top = 0
Left = 46 Top = 0 Action = OpenProcedure end object CloseActiveButton: TToolButton Left = 69 Top = 0 Action = CloseActiveDataSet
Left = 46 Top = 0 Action = OpenProcedure end object CloseActiveButton: TToolButton Left = 69 Top = 0 Action = CloseActiveDataSet end
Left = 46 Top = 0 Action = OpenProcedure end object CloseActiveButton: TToolButton Left = 69 Top = 0 Action = CloseActiveDataSet end object Sep1: TToolButton Left = 92
Left = 46 Top = 0 Action = OpenProcedure end object CloseActiveButton: TToolButton Left = 69 Top = 0 Action = CloseActiveDataSet end object Sep1: TToolButton
Left = 46 Top = 0 Action = OpenProcedure end object CloseActiveButton: TToolButton Left = 69 Top = 0 Action = CloseActiveDataSet end object Sep1: TToolButton Left = 92 Top = 0 Width = 14
Left = 46 Top = 0 Action = OpenProcedure end object CloseActiveButton: TToolButton Left = 69 Top = 0 Action = CloseActiveDataSet end object Sep1: TToolButton Left = 92 Top = 0 Width = 14 Caption = 'Sep1'
Left = 46 Top = 0 Action = OpenProcedure end object CloseActiveButton: TToolButton Left = 69 Top = 0 Action = CloseActiveDataSet end object Sep1: TToolButton Left = 92 Top = 0 Width = 14 Caption = 'Sep1' ImageIndex = 11
Left = 46 Top = 0 Action = OpenProcedure end object CloseActiveButton: TToolButton Left = 69 Top = 0 Action = CloseActiveDataSet end object Sep1: TToolButton Left = 92 Top = 0 Width = 14 Caption = 'Sep1' ImageIndex = 11 Style = tbsSeparator
Left = 46 Top = 0 Action = OpenProcedure end object CloseActiveButton: TToolButton Left = 69 Top = 0 Action = CloseActiveDataSet end object Sep1: TToolButton Left = 92 Top = 0 Width = 14 Caption = 'Sep1' ImageIndex = 11 Style = tbsSeparator end
Left = 46 Top = 0 Action = OpenProcedure end object CloseActiveButton: TToolButton Left = 69 Top = 0 Action = CloseActiveDataSet end object Sep1: TToolButton Left = 92 Top = 0 Width = 14 Caption = 'Sep1' ImageIndex = 11 Style = tbsSeparator end object LoadPacketButton:
Left = 46 Top = 0 Action = OpenProcedure end object CloseActiveButton: TToolButton Left = 69 Top = 0 Action = CloseActiveDataSet end object Sep1: TToolButton Left = 92 Top = 0 Width = 14 Caption = 'Sep1' ImageIndex = 11 Style = tbsSeparator end object LoadPacketButton: TToolButton
Left = 46 Top = 0 Action = OpenProcedure end object CloseActiveButton: TToolButton Left = 69 Top = 0 Action = CloseActiveDataSet end object Sep1: TToolButton Left = 92 Top = 0 Width = 14 Caption = 'Sep1' ImageIndex = 11 Style = tbsSeparator end object LoadPacketButton: TToolButton Left = 106
Left = 46 Top = 0 Action = OpenProcedure end object CloseActiveButton: TToolButton Left = 69 Top = 0 Action = CloseActiveDataSet end object Sep1: TToolButton Left = 92 Top = 0 Width = 14 Caption = 'Sep1' ImageIndex = 11 Style = tbsSeparator end object LoadPacketButton: TToolButton

7
end
object OpenDialog: TOpenDialog
DefaultExt = 'dtg'
Filter =
'TableGram Files
(*.dtg) *.dtg Delphi form
(*.dfm) *.dfm All File' +
's (*.*) *.*'
Title = 'Load Data File'
Left = 627
Top = 57
end
object SaveDialog: TSaveDialog
DefaultExt = 'dtg'
Filter =
'TableGram Files
(*.dtg) *.dtg Delphi form
(*.dfm) *.dfm All File' +
's (*.*) *.*'
Options = [ofOverwritePrompt]
Title = 'Save Data File'
Left = 459
Top = 371
end
object PopupMenu1: TPopupMenu
<b>j</b>
OnPopup = PopupMenu1Popup
OnPopup = PopupMenu1Popup
OnPopup = PopupMenu1Popup Left = 544
OnPopup = PopupMenu1Popup Left = 544 Top = 115 object AssignField1: TMenuItem
OnPopup = PopupMenu1Popup Left = 544 Top = 115 object AssignField1: TMenuItem Caption = '≔ Field'
OnPopup = PopupMenu1Popup Left = 544 Top = 115 object AssignField1: TMenuItem
OnPopup = PopupMenu1Popup Left = 544 Top = 115 object AssignField1: TMenuItem Caption = '≔ Field' OnClick = FieldSelect
OnPopup = PopupMenu1Popup Left = 544 Top = 115 object AssignField1: TMenuItem Caption = '≔ Field' OnClick = FieldSelect end end
OnPopup = PopupMenu1Popup Left = 544 Top = 115 object AssignField1: TMenuItem Caption = '≔ Field' OnClick = FieldSelect end
OnPopup = PopupMenu1Popup Left = 544 Top = 115 object AssignField1: TMenuItem Caption = '≔ Field' OnClick = FieldSelect end end object DetailDataSource: TDataSource
OnPopup = PopupMenu1Popup Left = 544 Top = 115 object AssignField1: TMenuItem Caption = '≔ Field' OnClick = FieldSelect end end object DetailDataSource: TDataSource OnStateChange =
OnPopup = PopupMenu1Popup Left = 544 Top = 115 object AssignField1: TMenuItem Caption = '≔ Field' OnClick = FieldSelect end end object DetailDataSource: TDataSource OnStateChange = DataSourceStateChange
OnPopup = PopupMenu1Popup Left = 544 Top = 115 object AssignField1: TMenuItem Caption = '≔ Field' OnClick = FieldSelect end end object DetailDataSource: TDataSource OnStateChange = DataSourceStateChange OnDataChange =
OnPopup = PopupMenu1Popup Left = 544 Top = 115 object AssignField1: TMenuItem Caption = '≔ Field' OnClick = FieldSelect end end object DetailDataSource: TDataSource OnStateChange = DataSourceStateChange
OnPopup = PopupMenu1Popup Left = 544 Top = 115 object AssignField1: TMenuItem Caption = '≔ Field' OnClick = FieldSelect end end object DetailDataSource: TDataSource OnStateChange = DataSourceStateChange OnDataChange = DataSourceDataChange
OnPopup = PopupMenu1Popup Left = 544 Top = 115 object AssignField1: TMenuItem Caption = '≔ Field' OnClick = FieldSelect end end object DetailDataSource: TDataSource OnStateChange = DataSourceStateChange OnDataChange = DataSourceDataChange OnUpdateData =
OnPopup = PopupMenu1Popup Left = 544 Top = 115 object AssignField1: TMenuItem Caption = '≔ Field' OnClick = FieldSelect end end object DetailDataSource: TDataSource OnStateChange = DataSourceStateChange OnDataChange = DataSourceDataChange OnUpdateData = DataSourceUpdateData Left = 19
OnPopup = PopupMenu1Popup Left = 544 Top = 115 object AssignField1: TMenuItem Caption = '≔ Field' OnClick = FieldSelect end end object DetailDataSource: TDataSource OnStateChange = DataSourceStateChange OnDataChange = DataSourceDataChange OnUpdateData = DataSourceUpdateData
OnPopup = PopupMenu1Popup Left = 544 Top = 115 object AssignField1: TMenuItem Caption = '≔ Field' OnClick = FieldSelect end end object DetailDataSource: TDataSource OnStateChange = DataSourceStateChange OnDataChange = DataSourceDataChange OnUpdateData = DataSourceUpdateData Left = 19 Top = 393 end
OnPopup = PopupMenu1Popup Left = 544 Top = 115 object AssignField1: TMenuItem Caption = '≔ Field' OnClick = FieldSelect end end object DetailDataSource: TDataSource OnStateChange = DataSourceStateChange OnDataChange = DataSourceDataChange OnUpdateData = DataSourceUpdateData Left = 19 Top = 393 end object MainMenu1: TMainMenu
OnPopup = PopupMenu1Popup Left = 544 Top = 115 object AssignField1: TMenuItem Caption = '≔ Field' OnClick = FieldSelect end end object DetailDataSource: TDataSource OnStateChange = DataSourceStateChange OnDataChange = DataSourceDataChange OnUpdateData = DataSourceUpdateData Left = 19 Top = 393 end object MainMenu1: TMainMenu Left = 529
OnPopup = PopupMenu1Popup Left = 544 Top = 115 object AssignField1: TMenuItem Caption = '≔ Field' OnClick = FieldSelect end end object DetailDataSource: TDataSource OnStateChange = DataSourceStateChange OnDataChange = DataSourceDataChange OnUpdateData = DataSourceUpdateData Left = 19 Top = 393 end object MainMenu1: TMainMenu

end
object
MidasApplyUpdatesButton:
TToolButton
Left = 263
Top = 0
Action = MidasApplyUpdates
end
object MidasCancelButton:
TToolButton
Left = 286
Top = 0
Action = MidasCancelUpdates
end
object ToolButton3: TToolButton
Left = 309
Top = 0
Width = 14
Caption = 'ToolButton3'
ImageIndex = 9
Style = tbsSeparator
end
object ClearEventsButton:
TToolButton
Left = 323
Top = 0
Action = ClearEventLog
end
end
object ProgressBar: TProgressBar
Left = 0
Top = 438
Width = 301
Height = 22
Step = 1
TabOrder = 5
Visible = False
end
object MasterDataSource:
TDataSource
OnStateChange =
DataSourceStateChange
OnDataChange =
DataSourceDataChange
OnUpdateData =
DataSourceUpdateData
Left = 19
Top = 283

end
object N1: TMenuItem
Caption = '-'
end
object Saveformtofile1:
TMenuItem
Action = StreamFormOut
end
object LoadFormfromFile1:
TMenuItem
Action = StreamFormIn
end
object N2: TMenuItem
Caption = '-'
end
object Exit1: TMenuItem
Action = ExitApplication
end
end
object EditMenu: TMenuItem
Caption = '&Edit'
object BatchUpdates1:
TMenuItem
Action = BatchUpdate
end
object CancelBatch1:
TMenuItem
Action = CancelBatch
end
object N6: TMenuItem
Caption = '-'
end
object ApplyUpdatesMidas1:
TMenuItem
Action = MidasApplyUpdates
end
object CancelUpdatesMidas1:
TMenuItem
Action = MidasCancelUpdates
end
object N7: TMenuItem
Caption = '-'
end
object ClearField1: TMenuItem
Action = ClearField
end
object RefreshData1:
TMenuItem

Caption = '&File'
OnClick = FileMenuClick
object OpenTable1: TMenuItem
Action = OpenTable
end
object OpenQuery1: TMenuItem
Action = OpenQuery
end
object OpenProcedure1:
TMenuItem
Action = OpenProcedure
end
object FileReopen: TMenuItem
Caption = 'Reopen'
object ReopenHolder:
TMenuItem
Caption = 'PlaceHolder'
end
end
object ExecuteQuery1:
TMenuItem
Action = ExecuteCommand
end
object CloseActiveDataset1:
TMenuItem
Action = CloseActiveDataSet
end
object DisconnectDataSet1:
TMenuItem
Action = DisconnectDataSet
end
object Disconnect1: TMenuItem
Action = CloseConnection
end
object N3: TMenuItem
Caption = '-'
end
object LoadDatapacket1:
TMenuItem
Action = LoadFromFile
end
object SaveDataPacket1:
TMenuItem
Action = SaveToFile
end
object LoadBlobfromfile1:
TMenuItem
Action = LoadBlobFromFile

object CurTypeForwardOnly:
TMenuItem
Caption = '&Forward Only'
GroupIndex = 1
RadioItem = True
OnClick = RadioItemClick
end
object CurTypeKeyset:
TMenuItem
Caption = '&Keyset'
Checked = True
GroupIndex = 1
RadioItem = True
OnClick = RadioItemClick
end
object Dynamic1: TMenuItem
Caption = '&Dynamic'
GroupIndex = 1
RadioItem = True
OnClick = RadioItemClick
end
object CurTypeStatic:
TMenuItem
Caption = '&Static'
GroupIndex = 1
RadioItem = True
OnClick = RadioItemClick
end
end
object LockTypeItem:
TMenuItem
Caption = '&Lock Type'
object LckTypeUnspecified:
TMenuItem
Caption = '&Unspecified'
GroupIndex = 1
RadioItem = True
OnClick = RadioItemClick
end
object LckTypeReadOnly:
TMenuItem
Caption = '&Read Only'
GroupIndex = 1
RadioItem = True
OnClick = RadioItemClick
end
object LckTypePessimistic:
TMenuItem

1 4 D 6 1D 4
Action = RefreshData
end
end
object ViewMenu: TMenuItem
Caption = '&View'
object DisplayDetailTable1:
TMenuItem
Action = DisplayDetails
end
object BlobfieldasImage1:
TMenuItem
Action = BlobAsImage
end
object N5: TMenuItem
Caption = '-'
end
object ViewEvents1: TMenuItem
Action = ViewEvents
end
object ClearEventLog1:
TMenuItem
Action = ClearEventLog
end
end
object SettingsMenu: TMenuItem
Caption = '&Settings'
object UseClientCursorItem:
TMenuItem
Action = UseClientCursor
end
object UseadCmdTableDirect1:
TMenuItem
Action = UseTableDirect
end
object UseShapeProvider1:
TMenuItem
Action = UseShapeProvider
end
object CursorTypeItem:
TMenuItem
Caption = 'C&ursor Type'
object CurTypeUnspecified:
TMenuItem
Caption = '&Unspecified'
GroupIndex = 1
RadioItem = True
OnClick = RadioItemClick
end

Astion - Holm About
Action = HelpAbout
Caption = '&About'
end
end
end
object ActionList1: TActionList
Images = ImageList1
Left = 624
Top = 190
object BatchUpdate: TAction
Category = 'Edit'
Caption = '&Apply Batch
Updates'
Hint = 'Apply pending batch
updates to the server'
ImageIndex = 3
ShortCut = 16450
OnExecute =
BatchUpdateExecute
OnUpdate = EditActionsUpdate
end
object CancelBatch: TAction
Category = 'Edit'
Caption = '&Cancel Batch
Updates'
Hint = 'Cancel pending batch
updates'
ImageIndex = 14
OnExecute =
CancelBatchExecute
end
object OpenProcedure: TAction
Category = 'File'
Caption = 'Open & Procedure'
Hint = 'Open stored
procedure(s)'
ImageIndex = 12
ShortCut = 16464
OnExecute =
OpenProcedureExecute
end
object OpenTable: TAction
Category = 'File'
Caption = 'Open & Table'
Hint = 'Open Table(s)'
• • • • • • • • • • • • • • • • • • • •
ImageIndex = 0 ShortCut = 16468
OnExecute = OpenTableExecute

Caption = '&Pessimistic'
GroupIndex = 1
RadioItem = True
OnClick = RadioItemClick
end
object LckTypeOptimistic:
TMenuItem
Caption = '&Optimistic'
Checked = True
GroupIndex = 1
RadioItem = True
OnClick = RadioItemClick
end
object
•
LckTypeBatchOptimistic: TMenuItem
Caption = '&Batch Optimistic'
GroupIndex = 1
RadioItem = True
OnClick = RadioItemClick
end
end
object MaxRecords1:
TMenuItem
Action = MaxRecords
end
object EnableBCD1: TMenuItem
Action = EnableBCD
end
object N4: TMenuItem
Caption = '-'
end
object AsyncConnect1:
TMenuItem
Action = AsyncConnect
end
object AsyncExecute1:
TMenuItem
Action = AsyncExecute
end
object AsyncFetch1: TMenuItem
Action = AsyncFetch
end
end
object HelpMenu: TMenuItem
Caption = '&Help'
object HelpAboutItem:
TMenuItem

C / IFD I
Category = 'File'
Caption = '&Execute Command'
Hint = 'Execute the current
query'
ShortCut = 16453
OnExecute = ExecSQLExecute
end
object StreamFormOut: TAction
Category = 'File'
Caption = 'Stream & Form Out'
Hint = 'Stream for to a form file'
OnExecute =
StreamFormOutClick
end
object StreamFormIn: TAction
Category = 'File'
Caption = 'Stream Form &In'
-
Hint = 'Stream in form file'
OnExecute =
StreamFormInClick
end
object ClearField: TAction
Category = 'Edit'
Caption = '&Clear Field'
Hint = 'Set the current field to
Null'
ImageIndex = 6
ShortCut = 24643
OnExecute = ClearFieldExecute
end
object ViewEvents: TAction
Category = 'View'
Caption = '&Events'
Hint = 'Display event
information'
OnExecute = ViewEventsExecute
end
object DisplayDetails: TAction
Category = 'View'
Caption = '&Display Detail
Table'
Checked = True
Hint = 'Display detail or nested
table in the lower grid'
OnExecute =
<b>DisplayDetailsExecute</b>
end
object RefreshData: TAction
object ixii eshibata. I Attivii

end
object OpenQuery: TAction
Category = 'File'
Caption = 'Open & Query'
Hint = 'Open Query(s)'
ImageIndex = 1
<b>ShortCut = 16465</b>
OnExecute = OpenQueryExecute
end
object LoadFromFile: TAction
Category = 'File'
Caption = 'Load data from a
file'
Hint = 'Load data from a file'
ImageIndex = 10
ShortCut = 16460
OnExecute =
LoadFromFileExecute end
object CloseActiveDataSet:
TAction
Category = 'File'
Caption = '&Close Active
Dataset'
Hint = 'Close the active dataset'
ImageIndex = 2
OnExecute =
CloseActiveDataSetExecute
OnUpdate = FileActionsUpdate
end
object SaveToFile: TAction
Category = 'File'
Caption = '&Save data to a file'
Hint = 'Save data to a file'
ImageIndex = 11
ShortCut = 16467
OnExecute = SaveToFileExecute
end
object ExitApplication: TAction
Category = 'File'
Caption = 'E&xit'
Hint = 'Exit the application'
ShortCut = 32856
OnExecute =
ExitApplicationExecute end
object ExecuteCommand: TAction
1 ACUUII

OnExecute =
BooleanActionExecute
end
object UseShapeProvider:
TAction
Category = 'Settings'
Caption = 'Use &Shape
Provider'
Hint = 'Use the Shape Provider'
OnExecute =
UseShapeProviderExecute
end
object CloseConnection: TAction
Category = 'File'
Caption = '&Close Connection'
Hint = 'Close the current
connection'
OnExecute =
CloseConnectionExecute
end
object AsyncConnect: TAction
Category = 'Settings'
Caption = 'AsyncConnect'
Hint = 'Make the connection
asynchronously'
OnExecute =
BooleanActionExecute
end
object AsyncExecute: TAction
Category = 'Settings'
Caption = 'AsyncExecute'
Hint = 'Execute the command
asynchronously'
OnExecute =
BooleanActionExecute
end E. A. E. E. A. E. E. A. E.
object AsyncFetch: TAction
Category = 'Settings'
Caption = 'AsyncFetch'
Hint = 'Fetch records
asynchronously'
OnExecute =
BooleanActionExecute
end
object MaxRecords: TAction
Category = 'Settings'
Caption = '&Max Records'
Hint = 'Specify the max number

Category = 'Edit'
Caption = 'Re&fresh Data'
OnExecute =
RefreshDataExecute
end
object ClearEventLog: TAction
Category = 'View'
Caption = '&Clear Event Log'
Hint = 'Clear the event log'
ImageIndex = 7
OnExecute =
ClearEventLogExecute
OnUpdate =
ClearEventLogUpdate
end
object HelpAbout: TAction
Caption = 'HelpAbout'
Hint = 'Display About Box'
OnExecute = HelpAboutExecute
end
object PrevQuery: TAction
Category = 'File'
Hint = 'Display the previous
query from the history list'
OnExecute = PrevQueryExecute
OnUpdate = PrevQueryUpdate
end
object NextQuery: TAction
Category = 'File'
Hint = 'Display the next query
from the history list, or insert new
one.'
OnExecute = NextQueryExecute
end
object UseClientCursor: TAction
Category = 'Settings'
Caption = 'Use & Client Cursor'
Checked = True Hint = 'Use client side cursor'
OnExecute =
BooleanActionExecute
end
object UseTableDirect: TAction
Category = 'Settings'
Caption = 'Use
adCmd&TableDirect'
Hint = 'Use the
adCmdTableDirect command type'

Hint = 'View the current blob	of records in a dataset'
field in a DBImage'	OnExecute =
OnExecute =	MaxRecordsExecute
BlobAsImageExecute	end
OnUpdate =	object MidasApplyUpdates:
BlobAsImageUpdate	TAction
end	Category = 'Edit'
object LoadBlobFromFile:	Caption = 'Apply Updates
TAction	(Midas)'
Category = 'Edit'	Hint = 'Apply pending Midas
Caption = '&Load Blob from	updates'
file'	ImageIndex = 13
Hint = 'Load blob data from a	ShortCut = 16449
file'	OnExecute =
OnExecute =	MidasApplyUpdatesExecute
LoadBlobFromFileExecute	end
end	object MidasCancelUpdates:
end	TAction
object ImageList1: TImageList	Category = 'Edit'
Left = 347	Caption = 'Cancel Updates
Top = 366	(Midas)'
Bitmap = {	Hint = 'Cancel pending Midas
Divinup (	updates'
	ImageIndex = 9
00000}	OnExecute =
end	MidasCancelUpdatesExecute
object Connection:	end
TADOConnection	object EnableBCD: TAction
LoginPrompt = False	Category = 'Settings'
Provider =	Caption = 'EnableBCD'
'C:\Program Files\Common	Checked = True
Files\System\OLE DB\Data	Hint = 'Configure datasets to use
Links\DBDemos.u' +	BCD fields for numeric field types'
'dl'	OnExecute =
OnDisconnect =	BooleanActionExecute
ConnectionDisconnect	end
OnInfoMessage =	object DisconnectDataSet:
ConnectionInfoMessage	TAction
OnBeginTransComplete =	Category = 'File'
ConnectionBeginTransComplete	
OnCommitTransComplete =	Caption = '&Disconnect DataSet' Hint = 'Disconnect the dataset
ConnectionCommitTransComplete	fromt he connection'
OnRollbackTransComplete =	OnExecute =
ConnectionRollbackTransComplete	DisconnectDataSetExecute
OnConnectComplete =	
ConnectionConnectComplete	end
OnWillConnect =	object BlobAsImage: TAction
ConnectionWillConnect	Category = 'View'
a connection vy IIIC Offfice	Lantion = 'X, Righ field as Image'
OnExecuteComplete =	Caption = '&Blob field as Image'

D.C. Y.	
BeforeInsert =	ConnectionExecuteComplete
DataSetBeforeInsert	OnWillExecute =
AfterInsert = DataSetAfterInsert	ConnectionWillExecute
BeforeEdit = DataSetBeforeEdit	OnLogin = ConnectionLogin
AfterEdit = DataSetAfterEdit	Left = 541
BeforePost = DataSetBeforePost	Top = 162
AfterPost = DataSetAfterPost	end
BeforeCancel =	object MasterTable: TADOTable
<b>DataSetBeforeCancel</b>	Connection = Connection
AfterCancel =	BeforeOpen =
<b>DataSetAfterCancel</b>	MasterTableBeforeOpen
BeforeDelete =	AfterOpen =
<b>DataSetBeforeDelete</b>	MasterTableAfterOpen
AfterDelete = DataSetAfterDelete	BeforeClose =
BeforeScroll =	MasterTableBeforeClose
DataSetBeforeScroll	AfterClose = DataSetAfterClose
AfterScroll = DataSetAfterScroll	BeforeInsert =
OnDeleteError = DataSetError	DataSetBeforeInsert
OnEditError = DataSetError	AfterInsert = DataSetAfterInsert
OnNewRecord =	BeforeEdit = DataSetBeforeEdit
DataSetNewRecord	AfterEdit = DataSetAfterEdit
OnPostError = DataSetError	<b>BeforePost = DataSetBeforePost</b>
Left = 435	AfterPost = DataSetAfterPost
Top = 252	BeforeCancel =
end	<b>DataSetBeforeCancel</b>
object MasterQuery: TADOQuery	AfterCancel =
Connection = Connection	<b>DataSetAfterCancel</b>
BeforeOpen =	BeforeDelete =
MasterQueryBeforeOpen	DataSetBeforeDelete
AfterOpen =	AfterDelete = DataSetAfterDelete
MasterQueryAfterOpen	BeforeScroll =
BeforeClose =	DataSetBeforeScroll
MasterQueryBeforeClose	AfterScroll = DataSetAfterScroll
AfterClose = DataSetAfterClose	OnDeleteError = DataSetError
BeforeInsert =	OnEditError = DataSetError
DataSetBeforeInsert	OnNewRecord =
AfterInsert = DataSetAfterInsert	DataSetNewRecord
BeforeEdit = DataSetBeforeEdit	OnPostError = DataSetError
AfterEdit = DataSetAfterEdit	Left = 540
BeforePost = DataSetBeforePost	Top = 355
AfterPost = DataSetAfterPost	end
BeforeCancel =	object DetailTable: TADOTable
<b>DataSetBeforeCancel</b>	Connection = Connection
AfterCancel =	BeforeOpen =
<b>DataSetAfterCancel</b>	<b>DetailTableBeforeOpen</b>
BeforeDelete =	AfterOpen = DataSetAfterOpen
DataSetBeforeDelete	BeforeClose = DataSetBeforeClose
AfterDelete = DataSetAfterDelete	AfterClose = DataSetAfterClose

Defension -	DefensConell -
BeforeOpen =	BeforeScroll =
MasterProcBeforeOpen	DataSetBeforeScroll
AfterOpen =	AfterScroll = DataSetAfterScroll
MasterProcAfterOpen	OnDeleteError = DataSetError
BeforeClose = DataSetBeforeClose	OnEditError = DataSetError
AfterClose = DataSetAfterClose	OnNewRecord =
BeforeInsert =	DataSetNewRecord
<b>DataSetBeforeInsert</b>	OnPostError = DataSetError
AfterInsert = DataSetAfterInsert	Parameters = <>
BeforeEdit = DataSetBeforeEdit	Left = 622
AfterEdit = DataSetAfterEdit	Top = 123
BeforePost = DataSetBeforePost	end
AfterPost = DataSetAfterPost	object DetailQuery: TADOQuery
BeforeCancel =	Connection = Connection
<b>DataSetBeforeCancel</b>	BeforeOpen =
AfterCancel =	DetailQueryBeforeOpen
<b>DataSetAfterCancel</b>	AfterOpen = DataSetAfterOpen
BeforeDelete =	<b>BeforeClose = DataSetBeforeClose</b>
DataSetBeforeDelete	BeforeInsert =
AfterDelete = DataSetAfterDelete	<b>DataSetBeforeInsert</b>
BeforeScroll =	AfterInsert = DataSetAfterInsert
<b>DataSetBeforeScroll</b>	BeforeEdit = DataSetBeforeEdit
AfterScroll = DataSetAfterScroll	AfterEdit = DataSetAfterEdit
OnDeleteError = DataSetError	BeforePost = DataSetBeforePost
OnEditError = DataSetError	AfterPost = DataSetAfterPost
OnNewRecord =	BeforeCancel =
DataSetNewRecord	<b>DataSetBeforeCancel</b>
OnPostError = DataSetError	AfterCancel =
Parameters = <>	<b>DataSetAfterCancel</b>
Prepared = True	BeforeDelete =
Left = 543	DataSetBeforeDelete
Top = 219	AfterDelete = DataSetAfterDelete
end	BeforeScroll =
object ADODataSet:	<b>DataSetBeforeScroll</b>
TADODataSet	AfterScroll = DataSetAfterScroll
Connection = Connection	OnDeleteError = DataSetError
BeforeOpen = DataSetBeforeOpen	OnEditError = DataSetError
AfterOpen = DataSetAfterOpen	OnNewRecord =
BeforeClose = DataSetBeforeClose	DataSetNewRecord
AfterClose = DataSetAfterClose	OnPostError = DataSetError
BeforeInsert =	DataSource = DetailQuerySource
DataSetBeforeInsert	Parameters = <>
AfterInsert = DataSetAfterInsert	Left = 189
BeforeEdit = DataSetBeforeEdit	Top = 228
AfterEdit = DataSetAfterEdit	end
BeforePost = DataSetBeforePost	object MasterProc:
AfterPost = DataSetAfterPost	TADOStoredProc
	Connection = Connection
BeforeCancel =	Connection – Connection

end>
Left = 624
Top = 300
end
object Provider: TDataSetProvider
Left = 110
Top = 336
end
object DetailProc:
TADOStoredProc
Connection = Connection
BeforeOpen =
<b>DetailProcBeforeOpen</b>
AfterOpen = DataSetAfterOpen
BeforeClose = DataSetBeforeClose
AfterClose = DataSetAfterClose
BeforeInsert =
DataSetBeforeInsert
AfterInsert = DataSetAfterInsert
BeforeEdit = DataSetBeforeEdit
AfterEdit = DataSetAfterEdit
BeforePost = DataSetBeforePost
AfterPost = DataSetAfterPost
BeforeCancel =
DataSetBeforeCancel
AfterCancel =
DataSetAfterCancel
BeforeDelete =
DataSetBeforeDelete
AfterDelete = DataSetAfterDelete
BeforeScroll =
DataSetBeforeScroll
AfterScroll = DataSetAfterScroll
OnDeleteError = DataSetError
OnEditError = DataSetError
OnNewRecord =
DataSetNewRecord
OnPostError = DataSetError
DataSource = DetailProcSource
Parameters = <>
Prepared = True
Left = 360
Top = 252
end
object DetailProcSource:
TDataSource
DataSet = MasterProc
Left = 265

<b>DataSetBeforeCancel</b>
AfterCancel =
<b>DataSetAfterCancel</b>
BeforeDelete =
DataSetBeforeDelete
AfterDelete = DataSetAfterDelete
BeforeScroll =
DataSetBeforeScroll
AfterScroll = DataSetAfterScroll
OnDeleteError = DataSetError
OnEditError = DataSetError
OnNewRecord =
DataSetNewRecord
OnPostError = DataSetError
Parameters = <>
Left = 632
Top = 243
end
object DetailMasterSource:
TDataSource
DataSet = MasterTable
Left = 131
Top = 269
end
object DetailQuerySource:
TDataSource
DataSet = MasterQuery
Left = 110
Top = 221
end
object MasterClientData:
, ,
TClientDataSet
Aggregates = <>
Params = <>
ProviderName = 'Provider'
OnReconcileError =
MasterClientDataReconcileError
Left = 228
Top = 336
end
object ADOCommand:
TADOCommand
Connection = Connection
Prepared = True
Parameters = <
item
Size = -1
Value = Null

FieldName =
'NUMERIC_SCALE'
end
end
object DetailClientData:
TClientDataSet
Aggregates = <>
Params = <>
Left = 160
Top = 384
end
End
انتهى البرنامج والآن مع الصور والرسوم من
أعداد علاء الدين اللباد ٧٩٥٧٥١٤٩٠

الملف التنفيذي لبرنامج
(Actions)
عامدا
علاء الدين اللباد
واتف ۷۳۵۷۵۶۹۰
Actions
Procedure <b>S</b> 16
unit Main;
interface
uses
Windows, Messages, SysUtils,
Variants, Classes, Graphics,
Controls, Forms,
Dialogs, StdActns, ExtActns,
ActnList, ImgList, ComCtrls,
ToolWin,
StdCtrls, Menus, Buttons, ExtCtrls, jpeg, ListActns;

type

TForm1 = class(TForm)
ActionList1: TActionList;
ImageList1: TImageList;

RichEditBold1: TRichEditBold;

Top = 253
end
object FieldSchemaSource:
<b>TDataSource</b>
DataSet = FieldSchema
Left = 538
Top = 287
end
object FieldSchema: TADODataSet
Parameters = <>
Left = 617
Top = 366
object
FieldSchemaCOLUMN_NAME:
TWideStringField
DisplayLabel = 'FieldName'
DisplayWidth = 20
FieldName =
'COLUMN_NAME'
Size = 30
end
object FieldSchemaDATA TYPE:
TWordField
DisplayLabel = 'Type'
DisplayWidth = 20
FieldName = 'DATA_TYPE'
OnGetText =
FieldSchemaDATA_TYPEGetText
end
object
FieldSchemaCHARACTER_MAXI
MUM LENGTH: TIntegerField
DisplayLabel = 'Size'
FieldName =
'CHARACTER MAXIMUM LEN
GTH'
end
object
FieldSchemaNUMERIC PRECISIO
N: TWordField
DisplayLabel = 'Precision'
FieldName =
'NUMERIC PRECISION'
end
object
FieldSchemaNUMERIC_SCALE:
TSmallintField
DisplayLabel = 'Scale'

GroupedAction21: TMenuItem;
GroupedAction31: TMenuItem;
N1: TMenuItem;
AutoCheckAction1: TMenuItem;
FileRun1: TFileRun;
Label1: TLabel;
Run1: TMenuItem;
N2: TMenuItem;
FileOpen1: TFileOpen;
Open1: TMenuItem;
BrowseURL1: TBrowseURL;
DownLoadURL1:
TDownLoadURL;
SendMail1: TSendMail;
,
GroupedAction12: TMenuItem;
BrowseURL2: TMenuItem;
DownloadURL2: TMenuItem;
SendMail2: TMenuItem;
ColorSelect1: TColorSelect;
Dialog1: TMenuItem;
SelectColor1: TMenuItem;
OpenPicture1: TOpenPicture;
FontEdit1: TFontEdit;
PreviousTab1: TPreviousTab;
NextTab1: TNextTab;
Button3: TBitBtn;
Button4: TBitBtn;
SelectFont1: TMenuItem;
OpenPicture2: TMenuItem;
PageControl2: TPageControl;
ListTab: TTabSheet;
AutoCheckTab: TTabSheet;
DialogTab: TTabSheet;
FormatTab: TTabSheet;
EditTab: TTabSheet;
SearchTab: TTabSheet;
ToolBar2: TToolBar;
ToolButton18: TToolButton;
,
ToolButton19: TToolButton;
ToolButton20: TToolButton;
ToolButton21: TToolButton;
Memo2: TMemo;
Image1: TImage;
Panel1: TPanel;
SpeedButton1: TSpeedButton;
Button2: TBitBtn;
Button1: TBitBtn;
ToolBar1: TToolBar;

RichEditItalic1: TRichEditItalic;
RichEditUnderline1:
TRichEditUnderline;
RichEditStrikeOut1:
TRichEditStrikeOut;
RichEditBullets1:
TRichEditBullets;
RichEditAlignLeft1:
TRichEditAlignLeft;
RichEditAlignRight1:
TRichEditAlignRight;
RichEditAlignCenter1:
TRichEditAlignCenter;
SearchFind1: TSearchFind;
SearchFindNext1:
TSearchFindNext;
SearchReplace1: TSearchReplace;
SearchFindFirst1:
TSearchFindFirst;
GroupAction1: TAction;
GroupAction2: TAction;
GroupAction3: TAction;
AutoCheckAction: TAction;
StatusBar1: TStatusBar;
MainMenu1: TMainMenu;
File1: TMenuItem;
FileExit1: TFileExit;
Exit1: TMenuItem;
Search1: TMenuItem;
Find1: TMenuItem;
FindFirst1: TMenuItem;
FindNext1: TMenuItem;
Replace1: TMenuItem;
Edit1: TMenuItem;
EditCut1: TEditCut;
EditCopy1: TEditCopy;
EditPaste1: TEditPaste;
EditSelectAll1: TEditSelectAll;
EditUndo1: TEditUndo;
EditDelete1: TEditDelete;
Cut1: TMenuItem;
Copy1: TMenuItem;
Paste1: TMenuItem;
Delete1: TMenuItem;
SelectAll1: TMenuItem;
Undo1: TMenuItem;
AutoCheck1: TMenuItem;
GroupedAction11: TMenuItem;

RadioButton2: TRadioButton;
RadioButton3: TRadioButton;
ToolButton16: TToolButton;
ToolButton17: TToolButton;
ToolButton22: TToolButton;
ToolButton23: TToolButton;
ListBox1: TListBox;
ListBox2: TListBox;
SpeedButton4: TSpeedButton;
SpeedButton5: TSpeedButton;
SpeedButton6: TSpeedButton;
SpeedButton7: TSpeedButton;
SpeedButton8: TSpeedButton;
ListControlCopySelection1:
TListControlCopySelection;
ListControlDeleteSelection1:
TListControlDeleteSelection;
ListControlSelectAll1:
TListControlSelectAll;
ListControlClearSelection1:
TListControlClearSelection;
ListControlMoveSelection1:
TListControlMoveSelection;
Label8: TLabel;
Label9: TLabel;
StaticText1: TStaticText;
· · · · · · · · · · · · · · · · · · ·
StaticText2: TStaticText;
StaticText3: TStaticText;
Label11: TLabel;
Label7: TLabel;
Label10: TLabel;
ShortCutTab: TTabSheet;
Button7: TBitBtn;
ShortCutAction: TAction;
HotKey1: THotKey;
Button8: TBitBtn;
AddShortCut: TAction;
Label12: TLabel;
Label13: TLabel;
ShortCutList: TListBox;
Label14: TLabel;
FileOpenWith1: TFileOpenWith;
OpenWith1: TMenuItem;
FileTab: TTabSheet;
Label3: TLabel;
Button9: TBitBtn;
Button10: TBitBtn;
FileSaveAs1: TFileSaveAs;
,

ToolButton5: TToolButton; ToolButton7: TToolButton; ToolButton8: TToolButton; ToolButton11: TToolButton; ToolButton12: TToolButton; ToolButton12: TToolButton; ToolButton2: TToolButton; RichEdit1: TRichEdit; ToolBar3: TToolButton; ToolButton3: TToolButton; ToolButton4: TToolButton; ToolButton9: TToolButton; ToolButton10: TToolButton; ToolButton10: TToolButton; ToolButton13: TToolButton; ToolButton14: TToolButton; Memo3: TMemo; ListBoxTab: TTabSheet; CheckBox1: TCheckBox; StaticListAction1:
ToolButton7: TToolButton; ToolButton8: TToolButton; ToolButton11: TToolButton; ToolButton12: TToolButton; ToolButton1: TToolButton; ToolButton2: TToolButton; RichEdit1: TRichEdit; ToolBar3: TToolBar; ToolButton3: TToolButton; ToolButton4: TToolButton; ToolButton9: TToolButton; ToolButton10: TToolButton; ToolButton13: TToolButton; ToolButton14: TToolButton; Memo3: TMemo; ListBoxTab: TTabSheet; CheckBox1: TCheckBox;
ToolButton8: TToolButton; ToolButton11: TToolButton; ToolButton12: TToolButton; ToolButton1: TToolButton; ToolButton2: TToolButton; RichEdit1: TRichEdit; ToolBar3: TToolBar; ToolButton3: TToolButton; ToolButton4: TToolButton; ToolButton9: TToolButton; ToolButton10: TToolButton; ToolButton13: TToolButton; ToolButton14: TToolButton; Memo3: TMemo; ListBoxTab: TTabSheet; CheckBox2: TCheckBox; CheckBox1: TCheckBox;
ToolButton11: TToolButton; ToolButton12: TToolButton; ToolButton1: TToolButton; ToolButton2: TToolButton; RichEdit1: TRichEdit; ToolBar3: TToolBar; ToolButton3: TToolButton; ToolButton4: TToolButton; ToolButton9: TToolButton; ToolButton10: TToolButton; ToolButton13: TToolButton; ToolButton14: TToolButton; Memo3: TMemo; ListBoxTab: TTabSheet; CheckBox2: TCheckBox; CheckBox1: TCheckBox;
ToolButton12: TToolButton; ToolButton1: TToolButton; ToolButton2: TToolButton; RichEdit1: TRichEdit; ToolBar3: TToolBar; ToolButton3: TToolButton; ToolButton4: TToolButton; ToolButton9: TToolButton; ToolButton10: TToolButton; ToolButton13: TToolButton; ToolButton14: TToolButton; Memo3: TMemo; ListBoxTab: TTabSheet; CheckBox2: TCheckBox; CheckBox1: TCheckBox;
ToolButton1: TToolButton; ToolButton2: TToolButton; RichEdit1: TRichEdit; ToolBar3: TToolBar; ToolButton3: TToolButton; ToolButton4: TToolButton; ToolButton9: TToolButton; ToolButton10: TToolButton; ToolButton13: TToolButton; ToolButton14: TToolButton; Memo3: TMemo; ListBoxTab: TTabSheet; CheckBox2: TCheckBox; CheckBox1: TCheckBox;
ToolButton2: TToolButton; RichEdit1: TRichEdit; ToolBar3: TToolBar; ToolButton3: TToolButton; ToolButton4: TToolButton; ToolButton9: TToolButton; ToolButton10: TToolButton; ToolButton13: TToolButton; ToolButton14: TToolButton; Memo3: TMemo; ListBoxTab: TTabSheet; CheckBox2: TCheckBox; CheckBox1: TCheckBox;
ToolButton2: TToolButton; RichEdit1: TRichEdit; ToolBar3: TToolBar; ToolButton3: TToolButton; ToolButton4: TToolButton; ToolButton9: TToolButton; ToolButton10: TToolButton; ToolButton13: TToolButton; ToolButton14: TToolButton; Memo3: TMemo; ListBoxTab: TTabSheet; CheckBox2: TCheckBox; CheckBox1: TCheckBox;
ToolBar3: TToolBar; ToolButton3: TToolButton; ToolButton4: TToolButton; ToolButton9: TToolButton; ToolButton10: TToolButton; ToolButton13: TToolButton; ToolButton14: TToolButton; Memo3: TMemo; ListBoxTab: TTabSheet; CheckBox2: TCheckBox; CheckBox1: TCheckBox;
ToolBar3: TToolBar; ToolButton3: TToolButton; ToolButton4: TToolButton; ToolButton9: TToolButton; ToolButton10: TToolButton; ToolButton13: TToolButton; ToolButton14: TToolButton; Memo3: TMemo; ListBoxTab: TTabSheet; CheckBox2: TCheckBox; CheckBox1: TCheckBox;
ToolButton3: TToolButton; ToolButton4: TToolButton; ToolButton9: TToolButton; ToolButton10: TToolButton; ToolButton13: TToolButton; ToolButton14: TToolButton; Memo3: TMemo; ListBoxTab: TTabSheet; CheckBox2: TCheckBox; CheckBox1: TCheckBox;
ToolButton4: TToolButton; ToolButton9: TToolButton; ToolButton10: TToolButton; ToolButton13: TToolButton; ToolButton14: TToolButton; Memo3: TMemo; ListBoxTab: TTabSheet; CheckBox2: TCheckBox; CheckBox1: TCheckBox;
ToolButton9: TToolButton; ToolButton10: TToolButton; ToolButton13: TToolButton; ToolButton14: TToolButton; Memo3: TMemo; ListBoxTab: TTabSheet; CheckBox2: TCheckBox; CheckBox1: TCheckBox;
ToolButton10: TToolButton; ToolButton13: TToolButton; ToolButton14: TToolButton; Memo3: TMemo; ListBoxTab: TTabSheet; CheckBox2: TCheckBox; CheckBox1: TCheckBox;
ToolButton13: TToolButton; ToolButton14: TToolButton; Memo3: TMemo; ListBoxTab: TTabSheet; CheckBox2: TCheckBox; CheckBox1: TCheckBox;
ToolButton14: TToolButton; Memo3: TMemo; ListBoxTab: TTabSheet; CheckBox2: TCheckBox; CheckBox1: TCheckBox;
Memo3: TMemo; ListBoxTab: TTabSheet; CheckBox2: TCheckBox; CheckBox1: TCheckBox;
ListBoxTab: TTabSheet; CheckBox2: TCheckBox; CheckBox1: TCheckBox;
CheckBox2: TCheckBox; CheckBox1: TCheckBox;
CheckBox1: TCheckBox;
ŕ
Static Listaction 1.
TStaticListAction;
DeleteAction: TAction;
AddAction: TAction;
ClearAction: TAction;
ActiveAction: TAction;
SetIndexAction: TAction;
Label4: TLabel;
Label TLabel;
Labels: TLabel;
ListView1: TListView; ComboBoxEx1: TComboBoxEx;
AddBtn: TBitBtn;
,
AddEdit: TEdit;
DeleteBtn: TBitBtn;
DelEdit: TEdit;
Button5: TBitBtn;
IdxEdit: TEdit;
Button6: TBitBtn;
ActiveBtn: TBitBtn;
GroupedActions11: TMenuItem;
Previous1: TMenuItem;
Next1: TMenuItem;
ToolBar4: TToolBar;
ToolButton15: TToolButton;
SpeedButton2: TSpeedButton;
SpeedButton3: TSpeedButton;
RadioButton1: TRadioButton;

{ Private declarations }	FilePrintSetup1: TFilePrintSetup;
,	N3: TMenuItem;
<> procedure	,
UpdateShortCutList; public	PrintSetup1: TMenuItem;
{ Public declarations }	Memo1: TMemo;
,	Button11: TBitBtn;
end;	Button12: TBitBtn;
	LabeledEdit1: TLabeledEdit;
var	Button13: TBitBtn;
Form1: TForm1;	<> procedure
	FileOpen1Accept(Sender: TObject);
implementation	<> procedure
	FormCreate(Sender: TObject);
{\$R *.dfm}	<> procedure
	ColorSelect1Accept(Sender:
procedure	TObject);
TForm1.FileOpen1Accept(Sender:	<> procedure
TObject);	OpenPicture1Accept(Sender:
begin	TObject);
Label3.Caption :=	<> procedure
FileOpen1.Dialog.FileName;	FontEdit1Accept(Sender: TObject);
end;	<> procedure
	DeleteActionExecute(Sender:
procedure	TObject);
TForm1.FormCreate(Sender:	<> procedure
TObject);	AddActionExecute(Sender:
begin	TObject);
	<> procedure
FontEdit1.Dialog.Font.Assign(Font);	ClearActionExecute(Sender:
// Make sure the following actions	TObject);
are enabled even though they don't	<> procedure
do anything	ActiveActionExecute(Sender:
GroupAction1.DisableIfNoHandler	TObject);
:= False;	<> procedure
GroupAction2.DisableIfNoHandler	SetIndexActionExecute(Sender:
:= False;	TObject);
GroupAction3.DisableIfNoHandler	<> procedure
:= False;	ShortCutActionExecute(Sender:
	TObject);
AutoCheckAction.DisableIfNoHandl	<> procedure
er := False;	AddShortCutExecute(Sender:
UpdateShortCutList;	TObject);
end;	<> procedure
	FileSaveAs1Accept(Sender:
procedure	TObject);
TForm1.ColorSelect1Accept(Sender	<> procedure
: TObject);	SearchTabShow(Sender: TObject);
begin	<> procedure
Color :=	EditTabShow(Sender: TObject);
ColorSelect1.Dialog.Color;	private

StaticListAction1.Active;	end;
if StaticListAction1.Active then	enu,
ActiveBtn.Caption := 'Set	nwaaduwa
InActive'	procedure TForm 1 Open Pieture 1 Accent (Sondo
	TForm1.OpenPicture1Accept(Sende
else	r: TObject);
ActiveBtn.Caption := 'Set Active';	begin
end;	1 1 Di 4 1 IE EU 6
	Image1.Picture.LoadFromFile(Open
procedure	Picture1.Dialog.FileName);
TForm1.SetIndexActionExecute(Sen	end;
der: TObject);	
begin	procedure
StaticListAction1.ItemIndex :=	TForm1.FontEdit1Accept(Sender:
StrToInt(IdxEdit.Text);	TObject);
end;	begin
procedure	Font.Assign(FontEdit1.Dialog.Font);
TForm1.ShortCutActionExecute(Se	end;
nder: TObject);	
begin	procedure
ShowMessage('ShortCut action	TForm1.DeleteActionExecute(Sende
executed');	r: TObject);
end;	begin
procedure	StaticListAction1.Items.Delete(StrT
TForm1.UpdateShortCutList;	oInt(DelEdit.Text));
var	end;
I: Integer;	,
begin	procedure
ShortCutList.Items.BeginUpdate;	TForm1.AddActionExecute(Sender:
3 1	TObject);
try ShortCutList.Items.Clear;	begin
ShortCutList.Itellis.Clear;	with StaticListAction1.Items.Add
ShoutCutI ist Itams Add(ShoutCutT	do
ShortCutList.Items.Add(ShortCutT	Caption := AddEdit.Text;
oText(ShortCutAction.ShortCut));	end;
for I := 0 to	Ciiu,
ShortCutAction.SecondaryShortCut	nwaaduwa
s.Count - 1 do	procedure
	TForm1.ClearActionExecute(Sender
ShortCutList.Items.Add(ShortCutT	: TObject);
oText(ShortCutAction.SecondarySh	begin
ortCuts.ShortCuts[I]));	StaticListAction1.Items.Clear;
finally	end;
ShortCutList.Items.EndUpdate;	
end;	procedure
end;	TForm1.ActiveActionExecute(Sende
	r: TObject);
procedure	begin
	StaticListAction1.Active := not

1-<(OBJECT) Form1:
TForm1
Left = -4
Top = -4
BorderIcons = [biSystemMenu,
biMinimize]
BorderStyle = bsSingle
Caption = 'Action Demo'
ClientHeight = 692
ClientWidth = 1024
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
Menu = MainMenu1
OldCreateOrder = False
ShowHint = True
OnCreate = FormCreate
DesignSize = (
1024
692)
PixelsPerInch = 96
TextHeight = 13
2- <(OBJECT ) Label1:
TLabel
Left = 6
Top = 658
Width = 197
Height = 13
Anchors = [akLeft, akBottom]
Caption = 'Below is a StatusBar
with AutoHint = True'
end
<< (OBJECT ) Label8: TLabel
Left = 8
Top = 5
Width = 159
Height = 13
Caption = '(all of the menu items
use actions)'
end
////// (OBJECT ) Label11: TLabel
Left = 7
Top = 321

TForm1.AddShortCutExecute(Send
`
er: TObject);
begin
ShortCutAction.SecondaryShortCut s.Add(ShortCutToText(HotKey1.Ho tKey));
UpdateShortCutList;
end;
enu;
procedure TForm1.FileSaveAs1Accept(Sender: TObject); begin
Memo1.Lines.SaveToFile(FileSaveAs1.Dialog.FileName); end;
Cita,
procedure TForm1.SearchTabShow(Sender: TObject); begin Memo2.SetFocus; end;
procedure TForm1.EditTabShow(Sender: TObject); begin Memo3.SetFocus;
end;
end.

Actions
برنامج شامل بلغة البرمجة الدلفي
إعداد علاء الدين اللباد
.9 2 2 0 0 0 7 4 1
( ACTION )
الملف النصي : ١٨٩ (OBJECT)

DEEALH T CHARGET
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 2
end
////// (OBJECT ) PageControl2:
TPageControl
Left = 5
Top = 26
Width = 470
Height = 290
ActivePage = ShortCutTab
TabOrder = 3
////// (OBJECT ) FileTab:
TTabSheet
Caption = 'File'
ImageIndex = 8
////// (OBJECT ) Label3: TLabel
Left = 93
Top = 10
Width = 158
Height = 13
Caption = '(select File Open and
select a file)'
Font.Charset =
DEFAULT CHARSET
Font.Color = clRed
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
////// (OBJECT ) Button9: TBitBtn
Left = 6
Top = 5
Width = 75
Height = 25
Action = FileOpen1
Caption = '&Open'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'

Width = 259
Height = 13
Caption = 'Any control with a
Blue caption is connected to actions'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
////// (OBJECT ) StatusBar1:
TStatusBar
Left = 0
Top = 673 Width = 1024
WIUII — 1024
II.:-I.4 10
Height = 19
AutoHint = True
Panels = <>
end
////// (OBJECT ) Button3: TBitBtn
Left = 320
Top = 324
Width = 75
Height = 25
Action = PreviousTab1
Caption = '&Previous'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 1
end
////// (OBJECT ) Button4: TBitBtn
Left = 400
Top = 324
1 op = 324 Width = 75
Height = 25
Action = NextTab1
Caption = '&Next'
Font.Charset =

CI I D (
Glyph.Data = {
000}
end
////// (OBJECT ) Button12:
TBitBtn
Left = 134
Top = 65
Width = 75
Height = 25
Action = FileOpenWith1
Caption = 'Open &With'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 4
end
////// (OBJECT ) LabeledEdit1:
TLabeledEdit
Left = 219
Top = 67
Width = 161
Height = 21
EditLabel.Width = 156
EditLabel.Height = 13
EditLabel.Caption = 'Specify a
filename or leave blank'
TabOrder = 5
end
////// (OBJECT ) Button13:
TBitBtn
Left = 7
Top = 65
Width = 75
Height = 25
Action = FileRun1
Caption = '&Run' Font.Charset =
DEFAULT CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 6

Font.Style = []
ParentFont = False
TabOrder = 0
Glyph.Data = {
F7C}
end
////// (OBJECT ) Button10:
TBitBtn
Left = 6
Top = 35
Width = 75
Height = 25
Action = FilePrintSetup1
Caption = 'Print Set&up'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 1
end
////// (OBJECT ) Memo1: TMemo
Left = 89
Top = 132
Width = 366
Height = 123
Lines.Strings = (
'Memo1')
TabOrder = 2
end
////// (OBJECT ) Button11:
TBitBtn
Left = 6
Top = 132
Width = 75
Height = 25
Action = FileSaveAs1
Caption = 'Save &As'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 3
I WO O I WO I

1
end
////// (OBJECT ) ToolButton14:
TToolButton
Left = 285
Top = 2
Action = EditSelectAll1
end
end
////// (OBJECT ) Memo3: TMemo
Left = 0
Top = 57
Width = 462
Height = 205
Align = alClient
Lines.Strings = (
'Memo3')
TabOrder = 1
end
////// (OBJECT ) StaticText2:
TStaticText
Left = 0
Top = 40
Width = 462
Height = 17
Align = alTop
Caption = '(all of these buttons
use actions)'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 2
end
end
////// (OBJECT ) SearchTab:
TTabSheet
Caption = 'Search'
ImageIndex = 5
OnShow = SearchTabShow
////// (OBJECT ) ToolBar2:
TToolBar
Left = 0
Top = 0
Width = 462
Height = 40
11015111 10

end
end
////// (OBJECT )
TTabSheet
Caption = 'Edit'
ImageIndex = 4
OnShow = EditTabShow
////// (OBJECT ) ToolBar3:
TToolBar
Left = 0
Top = 0
Width = 462
Height = 40
ButtonHeight = 36
ButtonWidth = 57
Caption = 'ToolBar3'
Images = ImageList1
ShowCaptions = True
TabOrder = 0
////// (OBJECT ) ToolButton3:
TToolButton
Left = 0
Top = 2
Action = EditCut1
end
////// (OBJECT ) ToolButton4:
TToolButton
Left = 57
<b>Top = 2</b>
Action = EditCopy1
end
////// (OBJECT ) ToolButton9:
TToolButton
Left = 114
Top = 2
Action = EditPaste1
end
////// (OBJECT ) ToolButton10:
TToolButton
Left = 171
Top = 2
Action = EditDelete1
end
////// (OBJECT ) ToolButton13:
TToolButton
Left = 228
Top = 2
Action = EditUndo1

////// (OBJECT ) StaticText3:
TStaticText
Left = 0
Top = 40
Width = 462
Height = 17
Align = alTop
Caption = '(all of these buttons
use actions)'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 2
end
end
////// (OBJECT ) AutoCheckTab:
TTabSheet
Caption = 'AutoCheck'
ImageIndex = 1
////// (OBJECT ) SpeedButton2:
TSpeedButton
Left = 241
Top = 77
Width = 101
Height = 22
Action = GroupAction2
AllowAllUp = True
GroupIndex = 1
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
////// (OBJECT ) SpeedButton3:
TSpeedButton
Left = 242
Top = 51
<b>Width = 100</b>
Height = 22
Action = GroupAction1
AllowAllUp = True

ButtonHeight = 36
ButtonWidth = 58
Caption = 'ToolBar2'
Images = ImageList1
ShowCaptions = True
TabOrder = 0
////// (OBJECT ) ToolButton18:
TToolButton
Left = 0
Top = 2
Action = SearchFind1
end
////// (OBJECT ) ToolButton19:
TToolButton
Left = 58
Top = 2
Action = SearchFindNext1
end
////// (OBJECT ) ToolButton20:
TToolButton
Left = 116
Top = 2
Action = SearchReplace1
end
////// (OBJECT ) ToolButton21:
TToolButton
Left = 174
Top = 2
Action = SearchFindFirst1
end
end
////// (OBJECT ) Memo2: TMemo
Left = 0
Top = 57 $Width = 462$
Height = 205
Align = alClient
HideSelection = False
Lines.Strings = (
'Here is some sample text for
testing the search and replace acti' +
'ons. Simply click one of the '
'actions to find or replace text
within the memo.')
ScrollBars = ssVertical
TabOrder = 1
end

7
Font.Style = []
ParentFont = False
TabOrder = 0
end
////// (OBJECT ) CheckBox1:
TCheckBox
Left = 25
Top = 174
Width = 124
Height = 17
Action = AutoCheckAction
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 1
end
////// (OBJECT ) ToolBar4:
TToolBar
Left = 0
Top = 0
Width = 462
Height = 40
AutoSize = True
ButtonHeight = 36
ButtonWidth = 93
Caption = 'ToolBar4'
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
Images = ImageList1
ParentFont = False
ShowCaptions = True
TabOrder = 2
////// (OBJECT ) ToolButton15:
TToolButton
Left = 0
Top = 2
Action = AutoCheckAction
end
////// (OBJECT ) ToolButton16:
TToolButton

GroupIndex = 1
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
////// (OBJECT ) Label9: TLabel
Left = 8
Top = 243
Width = 259
Height = 13
Caption = '(all of the controls
on this tab are connected to actions)'
end
////// (OBJECT ) Label7: TLabel
·
Left = 9
Top = 51
Width = 170
Height = 112
AutoSize = False
Caption =
'The two checkboxes below are
both connected to the same action a'
+
'nd illustrate the use of the
AutoCheck proeprty of actions. Als'
+
'o notice the items on the
AutoCheck menu stay in sync as well
wi' +
'thout writting any code.'
WordWrap = True
end
////// (OBJECT ) CheckBox2:
TCheckBox
Left = 25
Top = 152
Width = 124
Height = 17
Action = AutoCheckAction
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'

////// (OBJECT ) RadioButton2:
TRadioButton
Left = 240
Top = 126
Width = 113
Height = 17
Action = GroupAction2
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 4
end
////// (OBJECT ) RadioButton3:
TRadioButton
Left = 240
Top = 145
Width = 113
Height = 17
Action = GroupAction3
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 5
end
end
////// (OBJECT ) ListTab:
TTabSheet
Caption = 'Static List'
////// (OBJECT ) Label4: TLabel
Left = 14
Top = 151
Width = 63
Height = 13
Caption = 'ComboBoxEx'
end
////// (OBJECT ) Label5: TLabel
Left = 193
Top = 145
Width = 39
Height = 13
Height – 13

Left = 93 Top = 2
WWY4 W.W. C
Width = 8
Caption = 'ToolButton16'
ImageIndex = 0
Style = tbsSeparator
end
////// (OBJECT ) ToolButton17:
TToolButton
Left = 101
Top = 2
Action = GroupAction1
Grouped = True
Style = tbsCheck
end
////// (OBJECT ) ToolButton22:
TToolButton
Left = 194
Top = 2
Action = GroupAction2
Grouped = True
Style = tbsCheck
end
////// (OBJECT ) ToolButton23:
TToolButton
Left = 287
<b>Top = 2</b>
Action = GroupAction3
Grouped = True
Style = tbsCheck
end
end
////// (OBJECT ) RadioButton1:
TRadioButton
Left = 240
Top = 107
Width = 113
Height = 17
Action = GroupAction1
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
ParentFont = False TabOrder = 3

Top = 10
Width = 107
Height = 25
Action = AddAction
Caption = '&Add Item'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 2
end
////// (OBJECT ) AddEdit: TEdit
Left = 129
Top = 12
Width = 121
Height = 21
TabOrder = 3
Text = 'New Item'
end
////// (OBJECT ) DeleteBtn:
TBitBtn
Left = 15
Top = 47 Width = 107
Height = 25
Action = DeleteAction
Caption = '&Delete Item at
Index'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 4
end
////// (OBJECT ) DelEdit: TEdit
Left = 129
Top = 47
Width = 121
Height = 21
TabOrder = 5
Text = '0'
end
VIII

Caption = 'ListView'
end
////// (OBJECT ) Label6: TLabel
Left = 132
Top = 112
Width = 201
Height = 29
AutoSize = False
Caption = 'The controls below
are connected to same
TStaticListAction'
WordWrap = True
end
////// (OBJECT ) ListView1:
TListView
Left = 190
Top = 163
Width = 143
Height = 94
Action = StaticListAction1
Columns = <
item
Width = -2
WidthType = (
-2)
end>
HideSelection = False
RowSelect = True
SmallImages = ImageList1
TabOrder = 0
ViewStyle = vsReport
end
////// (OBJECT ) ComboBoxEx1:
TComboBoxEx
Left = 16
Top = 169
Width = 128
Height = 22
Action = StaticListAction1
ItemHeight = 16
TabOrder = 1
Text = 'StaticListAction1'
Images = ImageList1
DropDownCount = 8
end
////// (OBJECT ) AddBtn:
TBitBtn
Left = 15
Latt - 13

Action = ActiveAction
Caption = 'Set InActive'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 9
end
end
////// (OBJECT ) DialogTab:
TTabSheet
Caption = 'Dialog'
ImageIndex = 2
////// (OBJECT ) Image1: TImage
Left = 0
Top = 28
Width = 462
Height = 234
Align = alClient
Proportional = True
end
////// (OBJECT ) Panel1: TPanel
Left = 0
Top = 0
Width = 462
Height = 28
Align = alTop
BevelOuter = byNone
TabOrder = 0
////// (OBJECT ) SpeedButton1:
TSpeedButton
Left = 2
Top = 2
Width = 83
Height = 22
Action = ColorSelect1
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlue
Font.Height = -11
Font. Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
////// (OBJECT ) Button2:

////// (OBJECT ) Button5:
TBitBtn
Left = 15
Top = 116
<b>Width = 107</b>
Height = 25
Action = ClearAction
Caption = '&Clear'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 6
end
////// (OBJECT ) IdxEdit: TEdit
Left = 129
Top = 83
Width = 121
Height = 21
TabOrder = 7
Text = '-1'
end
////// (OBJECT ) Button6:
TBitBtn
Left = 15
Top = 82
Width = 107
Height = 25
Action = SetIndexAction
Caption = 'Set &Index'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 8
end
////// (OBJECT ) ActiveBtn:
TBitBtn
Left = 267
Top = 10
Width = 75
Height = 25

G
Caption = 'ToolBar1'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
Images = ImageList1
ParentFont = False
ShowCaptions = True
TabOrder = 0
////// (OBJECT ) ToolButton5:
TToolButton
Left = 0
Top = 2
Action = RichEditBold1
end
////// (OBJECT ) ToolButton6:
TToolButton
Left = 64
Top = 2
Action = RichEditItalic1
end
////// (OBJECT ) ToolButton7:
TToolButton
Left = 128
Top = 2
Action = RichEditStrikeOut1
end
////// (OBJECT ) ToolButton8:
TToolButton
Left = 192
Top = 2 Action = RichEditUnderline1
Wrap = True
end
////// (OBJECT ) ToolButton11:
TToolButton
Left = 0
Top = 38
Action = RichEditBullets1
end
////// (OBJECT ) ToolButton12:
TToolButton
Left = 64
Top = 38
Action = RichEditAlignLeft1
end
CHU

TED: (D)
TBitBtn
Left = 93
<b>Top = 2</b>
Width = 83
Height = 22
Action = FontEdit1
Caption = 'Se&lect Font'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 0
end
////// (OBJECT ) Button1:
TBitBtn
Left = 183
Top = 2
Width = 87
Height = 22
Action = OpenPicture1
Caption = '&Open Picture'
Font.Charset = DEFAULT CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 1
end
end
end
////// (OBJECT ) FormatTab:
TTabSheet
Caption = 'Format'
ImageIndex = 3
////// (OBJECT ) ToolBar1:
TToolBar
Left = 0
Top = 0
Width = 462
Height = 76
AutoSize = True
ButtonHeight = 36
ButtonWidth = 64
Dutton Wintin UT

ImagaInday — 7
ImageIndex = 7
////// (OBJECT ) SpeedButton4:
TSpeedButton
Left = 164
Top = 21
Width = 90
Height = 19
Action =
ListControlClearSelection1
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
////// (OBJECT ) SpeedButton5:
TSpeedButton
Left = 164
<b>Top</b> = 44
<b>Width = 90</b>
Height = 19
Action =
ListControlCopySelection1
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
////// (OBJECT ) SpeedButton6:
TSpeedButton
Left = 164
Top = 68
Width = 90
Height = 19
Action =
ListControlDeleteSelection1
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False

////// (OBJECT ) ToolButton1:
TToolButton
Left = 128
Top = 38
Action =
RichEditAlignCenter1
end
////// (OBJECT ) ToolButton2:
TToolButton
Left = 192
Top = 38
Action = RichEditAlignRight1
end
end
////// (OBJECT ) RichEdit1:
TRichEdit
Left = 0
Top = 93
Width = 462
Height = 169
Align = alClient
HideSelection = False
Lines.Strings = (
'RichEdit1')
TabOrder = 1
end
////// (OBJECT ) StaticText1:
TStaticText
Left = 0
Top = 76
Width = 462
Height = 17
Align = alTop
Caption = '(all of these buttons
use actions)'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 2
end
end
////// (OBJECT ) ListBoxTab:
TTabSheet
Caption = 'List'

Items.Strings = (
'one'
'two'
'three'
'four'
'five'
'six'
'seven'
'eight'
'nine'
'ten')
MultiSelect = True
TabOrder = 0
end
////// (OBJECT ) ListBox2:
TListBox
Left = 260
Top = 21
Width = 154
Height = 151
ItemHeight = 13
TabOrder = 1
end
end
////// (OBJECT ) ShortCutTab:
TTabSheet
Caption = 'ShortCuts'
ImageIndex = 7
////// (OBJECT ) Label12: TLabel
Left = 126
Top = 41
Width = 221
Height = 13
Caption = 'This action responds
to the following shortcuts:'
end
////// (OBJECT ) Label13: TLabel
Left = 23
Top = 139
Width = 330
Height = 30
AutoSize = False
Caption =
'Type a shortcut key and click
the Add ShortCut button to add a n'
+
'ew shortcut to the
ShortCutAction'

end
////// (OBJECT ) SpeedButton7:
TSpeedButton
Left = 164
Top = 91
Width = 90
Height = 19
Action =
ListControlMoveSelection1
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
////// (OBJECT ) SpeedButton8:
TSpeedButton
Left = 164
Top = 115
Width = 90
Height = 19
Action = ListControlSelectAll1
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
////// (OBJECT ) Label10: TLabel
Left = 20
<b>Top</b> = 4
Width = 298
Height = 13
Caption = 'These list actions
work on listboxes, listviews and
comboboxes.'
end
////// (OBJECT ) ListBox1:
TListBox
Left = 18
Top = 21
Width = 138
Height = 151
ItemHeight = 13
itemineight 15

Continu IA II Chant Conti
Caption = 'Add ShortCut'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 2
end
////// (OBJECT ) ShortCutList:
TListBox
Left = 126
Top = 58
Width = 121
Height = 69
ItemHeight = 13
TabOrder = 3
end
end
end
////// (OBJECT ) ActionList1:
TActionList
Images = ImageList1
Left = 425
Top = 104
////// (OBJECT ) RichEditBold1:
TRichEditBold
Category = 'Format'
AutoCheck = True
Caption = '&Bold'
Hint = 'Bold'
ImageIndex = 0 ShortCut = 16450
end
////// (OBJECT ) RichEditItalic1:
TRichEditItalic
Category = 'Format'
AutoCheck = True
Caption = '&Italic'
Hint = 'Italic'
ImageIndex = 1
ShortCut = 16457
end
////// (OBJECT )
RichEditUnderline1:
TRichEditUnderline
Category = 'Format'

FocusControl = HotKey1
WordWrap = True
end
////// (OBJECT ) Label14: TLabel
Left = 17
<b>Top</b> = 7
Width = 321
Height = 13
Caption =  'This tab illustrates the new
SecondaryShortCuts property of
TAct' +
'ion:'
end
////// (OBJECT ) Button7:
TBitBtn
Left = 19
Top = 59
Width = 98
Height = 25
Action = ShortCutAction
Caption = 'ShortCut Action'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlue
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 0
end
////// (OBJECT ) HotKey1:
THotKey
Left = 22
Top = 174
Width = 121
Height = 19
o .
HotKey = 49217
Modifiers = [hkCtrl, hkAlt]
TabOrder = 1
end
////// (OBJECT ) Button8:
TBitBtn
Left = 21
Top = 200
Width = 75
Height = 25
Action = AddShortCut
11ction 11ddyllol (Cut

AutoCheck = True	
Caption = '&Center'	
Hint = 'Center Centers text	
between margins'	
ImageIndex = 7	
end	e
////// (OBJECT ) SearchFind1:	///
TSearchFind	Ric
Category = 'Search'	TR
Caption = '&Find'	
Dialog.Options = [frDown,	
frFindNext, frHideUpDown	
Hint = 'Find Finds the specified	
text'	
ImageIndex = 8	
ShortCut = 16454	<u>e</u>
	/// TD:
end	TR
////// (OBJECT)	
SearchFindNext1: TSearchFindNext	
Category = 'Search'	
Caption = 'Find &Next'	
Hint = 'Find Next Repeats the	the
last find'	
ImageIndex = 9	e
SearchFind = SearchFind1	///
ShortCut = 114	Ric
end	TR
////// (OBJECT ) SearchReplace1:	
TSearchReplace	
Category = 'Search'	
Caption = '&Replace'	
Hint = 'Replace Replaces specific	the
text with different text'	
ImageIndex = 10	e
end	///
////// (OBJECT )	Ric
SearchFindFirst1: TSearchFindFirst	TR
Category = 'Search'	110
Caption = 'F&ind First'	
Hint = 'Find First Finds the first	
occurance of specified text'	416.0
end	the
////// (OBJECT ) GroupAction1:	
TAction	e
Category = 'AutoCheck'	///
AutoCheck = True	Ric
Caption = 'Grouped Action 1'	TR
GroupIndex = 1	

A 4 CL 1 T
AutoCheck = True
Caption = '&Underline'
Hint = 'Underline'
ImageIndex = 2
ShortCut = 16469
end
////// (OBJECT )
RichEditStrikeOut1:
TRichEditStrikeOut
Category = 'Format'
AutoCheck = True
Caption = '&Strikeout'
Hint = 'Strikeout'
ImageIndex = 3
end
////// (OBJECT ) RichEditBullets1:
TRichEditBullets
Category = 'Format'
AutoCheck = True
Caption = '&Bullets'
Hint = 'Bullets Inserts a bullet on
the current line'
ImageIndex = 4
end
////// (OBJECT )
RichEditAlignLeft1:
TRichEditAlignLeft
Category = 'Format'
AutoCheck = True
Caption = 'Align &Left'
Hint = 'Align Left Aligns text at
the left indent'
ImageIndex = 5
end
////// (OBJECT )
RichEditAlignRight1:
TRichEditAlignRight
Category = 'Format'
AutoCheck = True
Caption = 'Align &Right'
Hint = 'Align Right Aligns text at
the right indent'
ImageIndex = 6
end
////// (OBJECT )
RichEditAlignCenter1:
TRichEditAlignCenter
Category = 'Format'

and	TT:4 —
end	Hint =
////// (OBJECT ) EditCut1:	'Grouped Action where only
TEditCut	one action can be checked at any
Category = 'Edit'	give ' +
Caption = 'Cu&t'	'time'
Hint = 'Cut Cuts the selection	end
and puts it on the Clipboard'	////// (OBJECT ) GroupAction2:
ImageIndex = 12	TAction
<b>ShortCut = 16472</b>	Category = 'AutoCheck'
end	AutoCheck = True
////// (OBJECT ) EditCopy1:	Caption = 'Grouped Action 2'
TEditCopy	GroupIndex = 1
Category = 'Edit'	Hint =
Caption = '&Copy'	'Grouped Action where only
Hint = 'Copy Copies the selection	one action can be checked at any
and puts it on the Clipboard'	give '+
ImageIndex = 13	'time'
ShortCut = 16451	end
end	////// (OBJECT ) GroupAction3:
////// (OBJECT ) EditPaste1:	TAction
TEditPaste	Category = 'AutoCheck'
Category = 'Edit'	AutoCheck = True
Caption = '&Paste'	Caption = 'Grouped Action 3'
Hint = 'Paste Inserts Clipboard	GroupIndex = 1
contents'	Hint =
ImageIndex = 14	'Grouped Action where only
ShortCut = 16470	one action can be checked at any
end	give '+
////// (OBJECT ) EditSelectAll1:	'time'
TEditSelectAll	end
	////// (OBJECT )
Category = 'Edit' Caption = 'Select &All'	AutoCheckAction: TAction
Hint = 'Select All Selects the	Category = 'AutoCheck'
entire document'	AutoCheck = True
ShortCut = 16449	Caption = 'AutoCheck Action'
	Hint =
end	'AutoCheck action: when it is
////// (OBJECT ) EditUndo1: TEditUndo	
	executed it will automatically set '+
Category = 'Edit'	'it''s checked state without
Caption = '&Undo'	having to write any code'
Hint = 'Undo Reverts the last	end
action'	////// (OBJECT ) FileExit1:
ImageIndex = 15	TFileExit
ShortCut = 16474	Category = 'File'
end	Caption = 'E&xit'
////// (OBJECT ) EditDelete1:	Hint = 'Exit Quits the
TEditDelete	application'
Category = 'Edit'	ImageIndex = 11

•,
item
Caption = 'five'
ImageIndex = 5
end
item
Caption = 'six'
end>
end
////// (OBJECT ) DeleteAction:
TAction
Category = 'Buttons'
Caption = '&Delete Item at
Index'
Hint = 'Delete the item indicated
from the StaticListAction'
OnExecute =
DeleteActionExecute
end
////// (OBJECT ) AddAction:
TAction
Category = 'Buttons'
Caption = '&Add Item'
Hint = 'Add the item to the
StaticListAction'
OnExecute = AddActionExecute
end
////// (OBJECT ) ClearAction:
TAction
Category = 'Buttons'
Caption = '&Clear'
Hint = 'Clear the
StaticListAction'
OnExecute =
ClearActionExecute
end
////// (OBJECT ) ActiveAction:
TAction
Category = 'Buttons'
Caption = 'Set InActive'
Hint = 'Toggle the Active
property of the StaticListAction'
OnExecute =
ActiveActionExecute
end
////// (OBJECT ) SetIndexAction:
TAction
Category = 'Buttons'
Caption = 'Set &Index'

Caption = '&Delete'
Hint = 'Delete Erases the
selection'
ImageIndex = 16
ShortCut = 46
end
////// (OBJECT ) FileRun1:
TFileRun
Category = 'File'
Browse = True
BrowseDlg.Filter =
'Programs *.exe;*.com;*.bat;*.cmd  All Files *.*'
BrowseDlg.Options =
[ofHideReadOnly, ofFileMustExist,
ofEnableSizing]
BrowseDlg.Title = 'Run'
Caption = '&Run'
Hint = 'Run Runs an application'
Operation = 'open'
ShowCmd = scShowNormal
end
////// (OBJECT )
StaticListAction1: TStaticListAction
Category = 'List'
Caption = 'StaticListAction1'
Images = ImageList1
Items = <
item
Caption = 'zero'
ImageIndex = 0
end
item
Caption = 'one'
ImageIndex = 1
end
item
Caption = 'two'
ImageIndex = 2
end
item
Caption = 'three'
ImageIndex = 3
end
item
Caption = 'four'
ImageIndex = 4
end
enu

Category = 'Dialog'	Hint = 'Set the selected item of
Caption = '&Open Picture'	the StaticListAction'
Hint = 'Open Picture'	OnExecute =
ShortCut = 16463	SetIndexActionExecute
OnAccept = OpenPicture1Accept	end
end	////// (OBJECT ) FileOpen1:
////// (OBJECT ) FontEdit1:	TFileOpen
TFontEdit	Category = 'File'
Category = 'Dialog'	Caption = '&Open'
Caption = 'Se&lect Font'	Hint = 'Open Opens an existing
Dialog.Font.Charset =	file'
DEFAULT CHARSET	ImageIndex = 17
Dialog.Font.Color =	ShortCut = 16463
clWindowText	OnAccept = FileOpen1Accept
Dialog.Font.Height = -11	end
Dialog.Font.Name = 'MS Sans	////// (OBJECT ) BrowseURL1:
Serif'	TBrowseURL
Dialog.Font.Style = []	Category = 'Internet'
Hint = 'Font Select'	Caption = '&Browse URL'
OnAccept = FontEdit1Accept	Hint = 'Browses to
end	http://www.borland.com'
////// (OBJECT ) PreviousTab1:	URL = 'http://www.borland.com'
TPreviousTab	end
	////// (OBJECT ) DownLoadURL1:
Category = 'Tab'	TDownLoadURL
TabControl = PageControl2	Category = 'Internet'
Caption = '&Previous'	Caption = '&Download URL'
Enabled = False	Hint = 'Downloads
Hint = 'Previous Go back to the	http://www.borland.com to a file
previous tab'	called ''test'''
end	
////// (OBJECT ) NextTab1:	URL = 'http://www.borland.com'
TNextTab	Filename = 'test'
Category = 'Tab'	end
TabControl = PageControl2	////// (OBJECT ) SendMail1:
Caption = '&Next'	TSendMail
Enabled = False	Category = 'Internet'
Hint = 'Next Go to the next tab'	Caption = '&Send Mail'
end	Hint = 'Send email'
////// (OBJECT )	end
ListControlCopySelection1:	////// (OBJECT ) ColorSelect1:
TListControlCopySelection	TColorSelect
Category = 'List'	Category = 'Dialog'
Caption = 'Copy >>'	Caption = 'Select &Color'
Destination = ListBox2	Hint = 'Color Select'
ListControl = ListBox1	OnAccept = ColorSelect1Accept
end	end
////// (OBJECT )	////// (OBJECT ) OpenPicture1:
ListControlDeleteSelection1:	TOpenPicture

Caption = 'Open &With'	TListControlDeleteSelection
Dialog.Filter = 'All Files *.*'	Category = 'List'
Dialog.Options =	Caption = 'Delete >>'
[ofHideReadOnly, ofFileMustExist,	ListControl = ListBox2
ofEnableSizing]	end
Dialog.Title = 'Open With'	////// (OBJECT )
end	ListControlSelectAll1:
////// (OBJECT ) FileSaveAs1:	TListControlSelectAll
TFileSaveAs	Category = 'List'
Category = 'File'	Caption = '<< Select All'
Caption = 'Save &As'	end
Dialog.Filter = 'Text File *.txt All	////// (OBJECT )
Files *.*'	ListControlClearSelection1:
Hint = 'Save As Saves the active	<b>TListControlClearSelection</b>
file with a new name'	Category = 'List'
ImageIndex = 18	Caption = 'Clear'
OnAccept = FileSaveAs1Accept	end
end	////// (OBJECT )
////// (OBJECT ) FilePrintSetup1:	ListControlMoveSelection1:
TFilePrintSetup	<b>TListControlMoveSelection</b>
Category = 'File'	Category = 'List'
Caption = 'Print Set&up'	Caption = 'Move >>'
Hint = 'Print Setup'	Destination = ListBox2
end	ListControl = ListBox1
end	end
////// (OBJECT ) ImageList1:	////// (OBJECT ) ShortCutAction:
TImageList	TAction
Left = 426	Category = 'ShortCut'
Top = 150	Caption = 'ShortCut Action'
Bitmap = {	ShortCut = 16466
	SecondaryShortCuts.Strings = (
0000000000000}	'Ctrl+Alt+W'
end	'Ctrl+Alt+X'
////// (OBJECT ) MainMenu1:	'Ctrl+Alt+Y'
TMainMenu	'Ctrl+Shift+Z')
Images = ImageList1	OnExecute =
Left = 424	<b>ShortCutActionExecute</b>
Top = 58	end
////// (OBJECT ) File1:	////// (OBJECT ) AddShortCut:
TMenuItem	TAction
Caption = '&File'	Category = 'ShortCut'
////// (OBJECT ) Open1:	Caption = 'Add ShortCut'
TMenuItem	OnExecute =
Action = FileOpen1	AddShortCutExecute
end	end
////// (OBJECT ) OpenWith1:	////// (OBJECT ) FileOpenWith1:
TMenuItem	<b>TFileOpenWith</b>
Action = FileOpenWith1	Category = 'File'

////// (OBJECT ) Search1:
TMenuItem
Caption = '&Search'
////// (OBJECT ) Find1:
TMenuItem
Action = SearchFind1
end
////// (OBJECT ) FindFirst1:
TMenuItem
Action = SearchFindFirst1
end
////// (OBJECT ) FindNext1: TMenuItem
Action = SearchFindNext1
end
////// (OBJECT ) Replace1:
TMenuItem
Action = SearchReplace1
end
end
////// (OBJECT ) AutoCheck1:
TMenuItem
Caption = '&AutoCheck'
////// (OBJECT )
GroupedAction11: TMenuItem
Action = GroupAction1
AutoCheck = True
end
////// (OBJECT )
GroupedAction21: TMenuItem
Action = GroupAction2
AutoCheck = True
end
////// (OBJECT )
GroupedAction31: TMenuItem
Action = GroupAction3
AutoCheck = True
end
0 = 1 0 2
////// (OBJECT ) N1: TMenuItem
Caption = '-'
end
////// (OBJECT )
AutoCheckAction1: TMenuItem
Action = AutoCheckAction
AutoCheck = True
end
end
////// (OBJECT )

end
////// (OBJECT ) N3: TMenuItem
Caption = '-'
end
////// (OBJECT ) PrintSetup1:
TMenuItem
Action = FilePrintSetup1
end
////// (OBJECT ) Run1:
TMenuItem
Action = FileRun1
end
////// (OBJECT ) N2: TMenuItem
Caption = '-'
end
////// (OBJECT ) Exit1:
TMenuItem
Action = FileExit1
end
end
////// (OBJECT ) Edit1:
TMenuItem
Caption = '&Edit'
////// (OBJECT ) Cut1:
TMenuItem
Action = EditCut1
end
////// (OBJECT ) Copy1:
TMenuItem
Action = EditCopy1
end
////// (OBJECT ) Paste1:
TMenuItem
Action = EditPaste1
end
////// (OBJECT ) Delete1:
TMenuItem
Action = EditDelete1
end
////// (OBJECT ) SelectAll1:
TMenuItem
Action = EditSelectAll1
end
////// (OBJECT ) Undo1:
TMenuItem
Action = EditUndo1
end
end
CIIU

هذا البرنامج يحتوي على أكثر من ١٠٠ تعليمة وموضوعا مختلفا وهو برنامج شامل لكل التعليمات البرمجية السابقة التي مرت خلال الدروس السابقة وعند قدرة الطالب على تشكيل هذا البرنامج بنفسه يصبح قادرا على وضع نفسه في أول طريق البرمجة وعندها نقول له مبروك فان طريق الألف ميل يبدأ بخطوة وأنت قد قطعت أكثر من ربع المسافة لكي تصبح مبرمجا وتكون قادرا على تشكيل أي برنامج بنفسك حسب المعطيات التي توصلت إليها وبهذا البرنامج تكون قد صنعت على الَّأْقُل ٢٠ برِنامجا فرعيا وأنا أؤمن بانٍ من يحفظ ألف بيت شعر يستطيع أن يكتب قصيدة وبالتالي فان من يحفظ ٢٠ برنامجا مختلفا يستطيع أن يكون

علاء الدين اللباد

◆988@Y@FV1

ALAEDDSN LUBBAD C.C.

GroupedAction12: TMenuItem
Caption = '&Internet'
////// (OBJECT ) BrowseURL2:
TMenuItem
Action = BrowseURL1
end
////// (OBJECT )
DownloadURL2: TMenuItem
Action = DownLoadURL1
end
////// (OBJECT ) SendMail2:
TMenuItem
Action = SendMail1
end
end
////// (OBJECT ) Dialog1:
TMenuItem
Caption = 'Dialog'
////// (OBJECT ) SelectColor1:
TMenuItem
Action = ColorSelect1
end
////// (OBJECT ) SelectFont1:
TMenuItem
Action = FontEdit1
end
////// (OBJECT ) OpenPicture2:
TMenuItem
Action = OpenPicture1
end
end
////// (OBJECT )
GroupedActions11: TMenuItem
AutoCheck = True
Caption = 'Tab'
Checked = True
////// (OBJECT ) Previous1:
TMenuItem
Action = PreviousTab1
end
////// (OBJECT ) Next1:
TMenuItem
Action = NextTab1
end
end
end
end <sup>1</sup>
VIIV

العلاء الدين اللباد للمحاسبة والبرمجيات ٩٤٤٥٧٥٣٧١

E 1)
Form1);
Application.Run;
end.
الواجهة الرئيسية ( الملف التنفيذي )
unit Main;
interface
uses
Windows, Messages, SysUtils,
Variants, Classes, Graphics,
Controls, Forms,
Dialogs, StdActns, ExtActns,
ActnList, ImgList, ComCtrls,
ToolWin,
StdCtrls, Menus, Buttons, ExtCtrls,
jpeg, ListActns;
<b>A</b>
type
TForm1 = class(TForm)
ActionList1: TActionList;
ImageList1: TImageList;
RichEditBold1: TRichEditBold;
RichEditItalic1: TRichEditItalic;
RichEditUnderline1:
TRichEditUnderline;
RichEditStrikeOut1:
TRichEditStrikeOut;
RichEditBullets1:
TRichEditBullets;
RichEditAlignLeft1:
TRichEditAlignLeft;
RichEditAlignRight1:
TRichEditAlignRight;
RichEditAlignCenter1:
TRichEditAlignCenter;
SearchFind1: TSearchFind;
SearchFindNext1:
TSearchFindNext;
,
SearchReplace1: TSearchReplace;
SearchFindFirst1:
TSearchFindFirst;
GroupAction1: TAction;
GroupAction2: TAction;
GroupAction3: TAction;
AutoCheckAction: TAction;
StatusBar1: TStatusBar;
MainMenu1: TMainMenu;

الملف التنفيذي للبرنامج
(program Actions)
ويحتوي على ٣٢ (procedure )
مختلفا بحيث يتعلم ألطالب أكثر
التعليمات البرمجية المتداولة في لغة
البرمجة الدلقي ٧
program Actions;
Uses
Forms,
Main in '\\ÇãËáÉ ãä ÇáÈÑäÇãÌ
ÏáÝí 7\Actions\Main.pas' {Form1};
{\$R *.res}
Dogin
Begin
Application.Initialize;
Application.CreateForm(TForm1,

D // 2 TED:/D/
Button3: TBitBtn;
Button4: TBitBtn;
SelectFont1: TMenuItem;
OpenPicture2: TMenuItem;
PageControl2: TPageControl;
ListTab: TTabSheet;
AutoCheckTab: TTabSheet;
DialogTab: TTabSheet;
FormatTab: TTabSheet;
EditTab: TTabSheet;
SearchTab: TTabSheet;
ToolBar2: TToolBar;
ToolButton18: TToolButton;
ToolButton19: TToolButton;
ToolButton20: TToolButton; ToolButton21: TToolButton;
Memo2: TMemo;
Image1: TImage;
Panel1: TPanel;
SpeedButton1: TSpeedButton;
Button2: TBitBtn;
Button1: TBitBtn;
ToolBar1: TToolBar;
ToolButton5: TToolButton;
ToolButton6: TToolButton;
ToolButton7: TToolButton;
ToolButton8: TToolButton;
ToolButton11: TToolButton;
ToolButton12: TToolButton;
ToolButton1: TToolButton;
ToolButton2: TToolButton;
RichEdit1: TRichEdit;
ToolBar3: TToolBar;
ToolButton3: TToolButton;
ToolButton4: TToolButton;
ToolButton9: TToolButton;
· · · · · · · · · · · · · · · · · · ·
ToolButton10: TToolButton;
ToolButton13: TToolButton;
ToolButton14: TToolButton;
Memo3: TMemo;
ListBoxTab: TTabSheet;
CheckBox2: TCheckBox;
CheckBox1: TCheckBox;
StaticListAction1:
TStaticListAction;
DeleteAction: TAction;
AddAction: TAction;
ClearAction: TAction;
·

File1: TMenuItem;
FileExit1: TFileExit;
Exit1: TMenuItem;
Search1: TMenuItem;
Find1: TMenuItem;
FindFirst1: TMenuItem;
FindNext1: TMenuItem;
Replace1: TMenuItem;
Edit1: TMenuItem;
EditCut1: TEditCut;
EditCopy1: TEditCopy;
EditPaste1: TEditPaste;
EditSelectAll1: TEditSelectAll;
EditUndo1: TEditUndo;
EditDelete1: TEditDelete;
Cut1: TMenuItem;
Copy1: TMenuItem;
Paste1: TMenuItem;
Delete1: TMenuItem;
SelectAll1: TMenuItem;
Undo1: TMenuItem;
AutoCheck1: TMenuItem;
GroupedAction11: TMenuItem;
GroupedAction21: TMenuItem;
GroupedAction31: TMenuItem;
N1: TMenuItem;
AutoCheckAction1: TMenuItem;
FileRun1: TFileRun;
Label1: TLabel;
Run1: TMenuItem;
N2: TMenuItem;
FileOpen1: TFileOpen;
Open1: TMenuItem;
BrowseURL1: TBrowseURL;
DownLoadURL1:
TDownLoadURL;
SendMail1: TSendMail;
GroupedAction12: TMenuItem;
BrowseURL2: TMenuItem;
DownloadURL2: TMenuItem;
SendMail2: TMenuItem;
ColorSelect1: TColorSelect;
Dialog1: TMenuItem;
SelectColor1: TMenuItem;
OpenPicture1: TOpenPicture;
FontEdit1: TFontEdit;
PreviousTab1: TPreviousTab;
NextTab1: TNextTab;

StaticText1: TStaticText;
StaticText2: TStaticText;
StaticText3: TStaticText;
Label11: TLabel;
Label7: TLabel;
Label10: TLabel;
ShortCutTab: TTabSheet;
Button7: TBitBtn;
ShortCutAction: TAction;
HotKey1: THotKey;
Button8: TBitBtn;
AddShortCut: TAction;
Label12: TLabel;
Label13: TLabel;
ShortCutList: TListBox;
Label14: TLabel;
FileOpenWith1: TFileOpenWith;
OpenWith1: TMenuItem;
FileTab: TTabSheet;
Label3: TLabel;
Button9: TBitBtn;
Button10: TBitBtn;
FileSaveAs1: TFileSaveAs;
FilePrintSetup1: TFilePrintSetup;
N3: TMenuItem;
PrintSetup1: TMenuItem;
Memo1: TMemo;
Button11: TBitBtn;
Button12: TBitBtn;
LabeledEdit1: TLabeledEdit;
Button13: TBitBtn;
ALALUBBAD1: TMenuItem;
ACC1: TMenuItem;
Action1: TAction;
Window1: TMenuItem;
Show1: TMenuItem;
Hide1: TMenuItem;
N4: TMenuItem;
ArrangeAll1: TMenuItem;
Cascade1: TMenuItem;
ile1: TMenuItem;
NewWindow1: TMenuItem;
(procedure)
FileOpen1Accept(Sender: TObject);
(procedure)
FormCreate(Sender: TObject);
(procedure)
ColorSelect1Accept(Sender:
~ ······························

ActiveAction: TAction;
SetIndexAction: TAction;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
ListView1: TListView;
ComboBoxEx1: TComboBoxEx;
AddBtn: TBitBtn;
AddEdit: TEdit;
DeleteBtn: TBitBtn;
DelEdit: TEdit;
Button5: TBitBtn;
IdxEdit: TEdit;
Button6: TBitBtn;
ActiveBtn: TBitBtn;
GroupedActions11: TMenuItem;
Previous1: TMenuItem;
Next1: TMenuItem;
ToolBar4: TToolBar;
ToolButton15: TToolButton;
SpeedButton2: TSpeedButton;
SpeedButton3: TSpeedButton;
RadioButton1: TRadioButton;
RadioButton2: TRadioButton;
RadioButton3: TRadioButton;
ToolButton16: TToolButton;
ToolButton17: TToolButton;
ToolButton22: TToolButton;
ToolButton23: TToolButton;
ListBox1: TListBox;
ListBox2: TListBox;
SpeedButton4: TSpeedButton;
SpeedButton5: TSpeedButton;
SpeedButton6: TSpeedButton;
SpeedButton7: TSpeedButton;
SpeedButton8: TSpeedButton;
ListControlCopySelection1:
TListControlCopySelection;
ListControlDeleteSelection1:
TListControlDeleteSelection;
ListControlSelectAll1:
TListControlSelectAll;
ListControlClearSelection1:
TListControlClearSelection;
ListControlMoveSelection1:
TListControlMoveSelection;
Label8: TLabel;
Label9: TLabel;

(procedure)	TObject);
TForm1.FileOpen1Accept(Sender:	()
TObject);	OpenPicture1Accept(Sender:
begin	TObject);
Label3.Caption :=	()
FileOpen1.Dialog.FileName;	FontEdit1Accept(Sender: TObject);
end;	()
	DeleteActionExecute(Sender:
(procedure)	TObject);
TForm1.FormCreate(Sender:	(procedure)
TObject);	AddActionExecute(Sender:
begin	TObject);
	(procedure)
FontEdit1.Dialog.Font.Assign(Font);	ClearActionExecute(Sender:
// Make sure the following actions	TObject);
are enabled even though they don't	(procedure)
do anything	ActiveActionExecute(Sender:
GroupAction1.DisableIfNoHandler	TObject);
:= False;	()
GroupAction2.DisableIfNoHandler	SetIndexActionExecute(Sender:
:= False;	TObject);
GroupAction3.DisableIfNoHandler	()
:= False;	ShortCutActionExecute(Sender:
	TObject);
AutoCheckAction.DisableIfNoHandl	(procedure)
er := False;	AddShortCutExecute(Sender:
UpdateShortCutList;	TObject);
end;	(procedure)
	FileSaveAs1Accept(Sender:
(procedure)	TObject);
TForm1.ColorSelect1Accept(Sender	(procedure)
: TObject);	SearchTabShow(Sender: TObject);
begin	(procedure)
Color :=	EditTabShow(Sender: TObject);
ColorSelect1.Dialog.Color;	private
end;	{ Private declarations }
	(procedure)
(procedure)	UpdateShortCutList;
TForm1.OpenPicture1Accept(Sende	public
r: TObject);	{ Public declarations }
begin	end;
Image1.Picture.LoadFromFile(Open	var
Picture1.Dialog.FileName);	Form1: TForm1;
end;	
	implementation
()	
TForm1.FontEdit1Accept(Sender:	{\$R *.dfm}
TObject):	

### <u>AC.&P</u>

end;	begin
(procedure)	Font.Assign(FontEdit1.Dialog.Font);
TForm1.ShortCutActionExecute(Se	end;
nder: TObject);	
begin	(procedure)
ShowMessage('ShortCut action	TForm1.DeleteActionExecute(Sende
executed');	r: TObject);
end;	begin
(procedure)	StaticListAction1.Items.Delete(StrT
TForm1.UpdateShortCutList;	oInt(DelEdit.Text));
var	end;
I: Integer;	
begin	(procedure)
ShortCutList.Items.BeginUpdate;	TForm1.AddActionExecute(Sender:
try	TObject);
ShortCutList.Items.Clear;	begin
,	with StaticListAction1.Items.Add
ShortCutList.Items.Add(ShortCutT	do
oText(ShortCutAction.ShortCut));	Caption := AddEdit.Text;
for I := 0 to	end;
ShortCutAction.SecondaryShortCut	
s.Count - 1 do	()
	TForm1.ClearActionExecute(Sender
ShortCutList.Items.Add(ShortCutT	: TObject);
oText(ShortCutAction.SecondarySh	begin
ortCuts.ShortCuts[I]));	StaticListAction1.Items.Clear;
finally	end;
ShortCutList.Items.EndUpdate;	
end;	()
end;	TForm1.ActiveActionExecute(Sende
	r: TObject);
(procedure)	begin
TForm1.AddShortCutExecute(Send	StaticListAction1.Active := not
er: TObject);	StaticListAction1.Active;
begin	if StaticListAction1.Active then
	ActiveBtn.Caption := 'Set
ShortCutAction.SecondaryShortCut	InActive'
s.Add(ShortCutToText(HotKey1.Ho	else
tKey));	ActiveBtn.Caption := 'Set Active';
UpdateShortCutList;	end;
end;	
	()
(procedure)	TForm1.SetIndexActionExecute(Sen
TForm1.FileSaveAs1Accept(Sender:	der: TObject);
TObject);	begin
begin	StaticListAction1.ItemIndex :=
	StrToInt(IdyEdit Text)

Memo1.Lines.SaveToFile(FileSaveA
s1.Dialog.FileName);
end;
(procedure)
TForm1.SearchTabShow(Sender:
TObject);
begin
Memo2.SetFocus;
end;
(procedure)
TForm1.EditTabShow(Sender:
TObject);
begin
Memo3.SetFocus;
end;
end.

انتهى البرنامج ونأمل أن تكونوا قد حققتم الفائدة المرجوة منه والله ولي التوفيق.

علاء الدين اللباد

• \$ \$ \$ @\@\\ \\

Caption = 'Marine Adventures'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -16
Font.Name = 'Times New
Roman'
Font.Style = [fsItalic,
fsUnderline]
ParentFont = False
end
< object \\ Bevel1: TBevel
Left = 10
Top = 150
Width = 147
Height = 2
Style = bsRaised
end
< object \\ Label1: TLabel
Left = 60
Top = 158
Width = 47
Height = 13
Caption = 'Loading'
end
< object \\ Image1: TImage
Left = 40
Top = 11
Width = 87
Height = 113
Picture.Data = {
1 ictui c.Data – {
end
end
End
< object \\ SearchDlg:
TSearchDlg
Left = 226
Top = 118 ActiveControl = DBGrid1
BorderIcons = [biSystemMenu]
BorderStyle = bsDialog
Caption = 'SearchDlg'
ClientHeight = 299
ClientWidth = 282
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET

<> Objects 10
الهلف النصي لبرناهج
(MastApp)
عاعدا
علاء الدين اللباد
هاتف ۷۳٬۹۷۵۹۹۰
الملف النصي للبرنامج اعداد
علاء الدين اللباد
.9 £ £ 0 Y 0 T Y 1
160 object
< object \\ SplashForm:
TSplashForm
Left = 605
Top = 209
ActiveControl = Panel1
BorderIcons = []
BorderStyle = bsNone
ClientHeight = 178
ClientWidth = 168
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = [] OldCreateOrder = True
Position = poScreenCenter PixelsPerInch = 96
TextHeight = 13
< object \\ Panel1: TPanel
Left = 0
Top = 0
Width = 168
Height = 178
Align = alClient
BevelInner = byLowered
Color = clSilver
TabOrder = 0
< object \\ Label3: TLabel
Left = 23
Top = 125
Width = 116
Height = 19

dgRowSelect]
TabOrder = 2
TitleFont.Charset =
DEFAULT_CHARSET
TitleFont.Color = clBlack
TitleFont.Height = -11
TitleFont.Name = 'MS Sans Serif'
TitleFont.Style = []
OnDblClick = DBGrid1DblClick
Columns = <
item
Expanded = False
FieldName = 'PartNo'
Visible = True
end
item
Expanded = False
FieldName = 'Description'
Width = 64
Visible = True
end
item
Expanded = False
FieldName = 'VendorNo'
Width = 64
Visible = True
end
item
Expanded = False
FieldName = 'OnHand'
Width = 64
Visible = True
end
item
Expanded = False
FieldName = 'OnOrder'
Visible = True
end
item
Expanded = False
FieldName = 'BackOrd'
Width = 64
Visible = True
end
item
Expanded = False
FieldName = 'Cost'
Width = 64

Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
PixelsPerInch = 96
TextHeight = 13
< object \\ Label1: TLabel
Left = 9
Top = 11
Width = 69
Height = 13
Alignment = taRightJustify
AutoSize = False
Caption = 'Search Field:'
end
< object \\ Label2: TLabel
Left = 7
Top = 40
Width = 71
Height = 13
Alignment = taRightJustify
AutoSize = False
Caption = 'Search Text:'
end
< object \\ SearchButton:
TSpeedButton
Left = 248
Top = 36
Width = 23
Height = 23
Hint = 'Search'
Enabled = False
Glyph.Data = {
NumGlyphs = 2
ParentShowHint = False
ShowHint = True
OnClick = SearchButtonClick
end
< object \\ DBGrid1: TDBGrid
Left = 8
Top = 66
Width = 264
Height = 181
DataSource = DataSource
Options = [dgTitles, dgIndicator,
dgColLines, dgRowLines,

- 1: 4WD 4 G
< object \\ DataSource:
TDataSource
DataSet = MastData.Parts
Left = 47
Top = 253
end
End
111111111111111111111111111111111111111
< object \\ QueryCustDlg:
TQueryCustDlg
Left = 372
Top = 297
ActiveControl = FromEdit
BorderIcons = [biSystemMenu]
BorderStyle = bsDialog
Caption = 'Specify Date Range'
ClientHeight = 96
ClientWidth = 318
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
V 13
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
< object \\ Bevel1: TBevel
Left = 5
Top = 6
Width = 225
Height = 84
Shape = bsFrame
end
< object \\ Label1: TLabel
Left = 12
Top = 34
Width = 23
Height = 13
Caption = '&From'
FocusControl = FromEdit
end
< object \\ Label2: TLabel
· object // Dabciz. I Dabci

Visible = True
end
item
Expanded = False
FieldName = 'ListPrice'
Visible = True
end>
end
< object \\ OKBtn: TButton
Left = 115
Top = 265
Width = 75
Height = 25
Caption = '&OK'
Default = True
ModalResult = 1
TabOrder = 3
end
< object \\ CancelBtn: TButton
Left = 199
Top = 265
Width = 75
Height = 25
Cancel = True
Caption = '&Cancel'
ModalResult = 2
TabOrder = 4
end
< object \\ SearchEd: TEdit
Left = 80
Top = 36
Width = 161
Height = 21
TabOrder = 1
OnChange = SearchEdChange
OnKeyPress = SearchEdKeyPress
end
< object \\ OrderCombo:
TComboBox
Left = 80
Top = 8
Width = 192
Height = 21
Style = csDropDownList
ItemHeight = 13
TabOrder = 0
OnChange = OrderComboChange
end

ShowHint = True
OnClick = PopupCalToBtnClick
end Evil (TEVIL)
< object \\ FromEdit: TEdit
Left = 46
Top = 31
Width = 147
Height = 21
TabOrder = 0
end
< object \\ ToEdit: TEdit
Left = 46
Top = 60
Width = 147
Height = 21
TabOrder = 1
end
< object \\ CancelBtn: TButton
Left = 240
Top = 34
<b>Width = 70</b>
Height = 25
Cancel = True
Caption = '&Cancel'
ModalResult = 2
TabOrder = 2
end
< object \\ OkBtn: TButton
Left = 240
Top = 6
Width = 70
Height = 25
Caption = '&OK'
Default = True
ModalResult = 1
TabOrder = 3
OnClick = OkBtnClick
end
End
\\\\\\
< object \\ PickRpt: TPickRpt
Left = 535
Top = 233
-
HelpContext = 5
BorderStyle = bsDialog
Caption = 'Report Selection'
ClientHeight = 106

<u>1</u>
Left = 12
Top = 63
Width = 13
Height = 13
Caption = '&To'
FocusControl = ToEdit
end
< object \\ Msglab: TLabel
Left = 12
Top = 11
Width = 197
Height = 16
AutoSize = False
Caption = 'Customers with
LastInvoiceDate ranging:'
WordWrap = True
end
< object \\ PopupCalBtnFrom:
TSpeedButton
Left = 197
Top = 31
<b>Width = 21</b>
Height = 21
Hint = 'Browse calendar'
Glyph.Data = {
333333000000333333333333333333
00000}
Layout = blGlyphRight
ParentShowHint = False
ShowHint = True
OnClick =
PopupCalBtnFromClick
end
< object \\ PopupCalToBtn:
<b>TSpeedButton</b>
Left = 197
Top = 60
Width = 21
Height = 21
Hint = 'Browse calendar'
Glyph.Data = {
, p (
333333000000333333333333333333
00000}
Layout = blGlyphRight
ParentShowHint = False
i ai viito ii villiit ii albu

Left = 151
Top = 40
Width = 73
Height = 25
Caption = '&View'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ModalResult = 1
ParentFont = False
TabOrder = 2
OnClick = ViewBtnClick
end
< object \\ ReportType:
TRadioGroup
Left = 9
Top = 6
Width = 129
Height = 87
Caption = '&Report'
ItemIndex = 0
Items.Strings = (
'C&ustomers Report'
'&Orders Report'
'&Invoice Report')
TabOrder = 3
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
< object \\ PickOrderNoDlg:
TPickOrderNoDlg
Left = 380
Top = 170
AutoScroll = False
Caption = 'Order No.'
ClientHeight = 190
ClientWidth = 166
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'

<u>-</u>
ClientWidth = 238
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
PixelsPerInch = 96
TextHeight = 13
< object \\ OKBtn: TButton
Left = 150
Top = 11
Width = 73
Height = 25
Caption = '&Print'
Default = True
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ModalResult = 1
ParentFont = False
TabOrder = 0
OnClick = OKBtnClick
end
< object \\ CloseBtn: TButton
Left = 151
Top = 69
Width = 73
Height = 25
Cancel = True
Caption = '&Cancel'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ModalResult = 2
ParentFont = False
TabOrder = 1
end
< object \\ ViewBtn: TButton
onject // viewdin. i duiton

	_
< object \\ OrdersByDateReport:	
TOrdersByDateReport	
Left = 0	
Top = 0	
Width = 816	
Height = 780	
Color = clBtnFace	
Font.Charset =	
DEFAULT_CHARSET	
Font.Color = clWindowText	Ι
Font.Height = -13	
Font.Name = 'Arial'	
Font.Style = []	
OldCreateOrder = True	
PixelsPerInch = 96	
TextHeight = 16	
End	
< object \\ MainForm:	
TMainForm	
Left = 224	
Top = 122	
HelpContext = 1	
ActiveControl = MainPanel	
BorderIcons = [biSystemMenu,	
biMinimize]	
BorderStyle = bsSingle	
Caption = 'Marine Adventures	
Order Entry'	
ClientHeight = 84	
ClientWidth = 401	]
Color = clBtnFace	
Font.Charset =	
DEFAULT_CHARSET	
Font.Color = clBlack	
Font.Height = -11	
Font.Name = 'MS Sans Serif'	
Font.Style = []	
Menu = MainMenu	N
OldCreateOrder = True	
OnCreate = FormCreate	
	F
OnDestroy = FormDestroy	

Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
OnShow = FormShow
PixelsPerInch = 96
TextHeight = 13
< object \\ Label1: TLabel
Left = 5
Top = 1
<b>Width = 94</b>
Height = 13
Caption = '&Select an Order No.'
FocusControl =
DBLookupListBox1
end
< object \\ CancelBtn: TButton
Left = 86
Top = 160
Width = 75
Height = 25
Cancel = True
Caption = '&Cancel'
ModalResult = 2
TabOrder = 0
end
< object \\ OKBtn: TButton
Left = 2
Top = 160
Width = 75
Height = 25
Caption = '&OK'
Default = True
ModalResult = 1
TabOrder = 1
end
< object \\ DBLookupListBox1:
TDBLookupListBox
Left = 5
Top = 16
Width = 153
Height = 134
KeyField = 'OrderNo'
ListField = 'OrderNo'
ListSource =
MastData.OrdersSource
TabOrder = 2
end
End

Caption = 'P&arts'
Glyph.Data = {
Layout = blGlyphTop
Spacing = 3
OnClick = BrowseParts
end
< object \\ ReportBtn:
TSpeedButton
Left = 196
Top = 1
Width = 66
Height = 45
Hint = 'Print custom reports'
Caption = '&Reports'
Glyph.Data = {
0000}
Layout = blGlyphTop
Spacing = 0
OnClick = ReportBtnClick
end
< object \\ HelpBtn: TSpeedButton
Left = 261
Top = 1
Width = 66
Height = 45
Hint = 'View technical help about
the application'
Caption = '&Help'
Glyph.Data = {
0303}
Layout = blGlyphTop
NumGlyphs = 2
Spacing = 0
OnClick = HelpBtnClick
end
< object \\ CloseBtn:
TSpeedButton
Left = 326
Top = 1
Width = 66
Height = 45
Hint = 'Close the application'
Caption = 'Cl&ose'
Glyph.Data = {
Layout = blGlyphTop
NumGlyphs = 2

TextHeight = 13
< object \\ MainPanel: TPanel
Left = 0
Top = 0
Width = 401
Height = 49
Align = alTop
ParentShowHint = False
ShowHint = True
TabOrder = 0
< object \\ OrderBtn:
TSpeedButton
Left = 1
Top = 1
Width = 66
Height = 45
Hint = 'Enter new order'
Caption = '&New Order'
Glyph.Data = {
333333333333000000}
Layout = blGlyphTop
Spacing = 0
OnClick = NewOrder
end
< object \\ BrowseBtn:
TSpeedButton
Left = 66
Top = 1
Width = 66
Height = 45
Hint = 'Browse and edit orders'
Caption = '&Browse'
Glyph.Data = {
S.J. P. Marie (
33333333333333365F0}
Layout = blGlyphTop
Spacing = 0 On Click = Provide Cust Ond
OnClick = BrowseCustOrd
end
< object \\ PartsBtn:
TSpeedButton
Left = 131
Top = 1
Width = 66
Height = 45
Hint = 'Browse and edit
inventory'

OnClick = CloseApp
end
end
< object \\ ViewMenu:
TMenuItem
Caption = '&View'
OnClick = ViewMenuClick
< object \\ ViewOrders:
TMenuItem
Caption = '&Orders'
OnClick = BrowseCustOrd
end
< object \\
ViewPartsInventory: TMenuItem
Caption = '&Parts/Inventory'
OnClick = BrowseParts
end
< object \\ N7: TMenuItem
Caption = '-'
end
< object \\ ViewStayOnTop:
TMenuItem
Caption = '&Stay On Top'
OnClick = ToggleStayonTop
end
< object \\ N1: TMenuItem
Caption = '-'
end
< object \\ ViewLocal:
TMenuItem
Caption = '&Local Data
(Paradox Data)'
Checked = True
GroupIndex = 1
RadioItem = True
OnClick = ViewLocalClick
end
< object \\ ViewRemote:
TMenuItem
Caption = '&Remote Data
(Local Interbase) '
GroupIndex = 1
RadioItem = True
OnClick = ViewRemoteClick
end
end
< object \\ HelpMenu:
TMenuItem
Caption = '&Help'

Spacing = 0
OnClick = CloseApp
end
end
< object \\ MainMenu:
TMainMenu
Left = 36
Top = 52
< object \\ FileMenu:
TMenuItem
Caption = '&File'
< object \\ FileNewOrder:
TMenuItem
Caption = '&New Order'
OnClick = NewOrder
end
< object \\ N5: TMenuItem
Caption = '-'
end
< object \\ FilePrintReport:
TMenuItem
Caption = 'Print &Report'
< object \\ PrintCustList:
TMenuItem
Caption = '&Customer List'
OnClick = CustomerReport
end
< object \\ PrintOrders:
TMenuItem
Caption = '&Order History'
OnClick = OrderReport
end
< object \\ PrintInvoice:
TMenuItem
Caption = '&Invoice'
OnClick = InvoiceReport
end
end
< object \\ FilePrinterSetup:
TMenuItem
Caption = 'Printer Setup'
OnClick = PrinterSetupClick
end
< object \\ N4: TMenuItem
Caption = '-'
end
< object \\ FileExit:
TMenuItem
Caption = 'E&xit'

PixelsPerInch = 96
TextHeight = 13
< object \\ Panel1: TPanel
Left = 0
Top = 0
Width = 313
Height = 36
Align = alTop
BevelOuter = bvNone
TabOrder = 0
< object \\ PrintBtn:
TSpeedButton
Left = 270
Top = 5
Width = 25
Height = 25
Hint = 'Print form image'
Glyph.Data = {
OnClick = PrintBtnClick
end
< object \\ Bevel1: TBevel
Left = 0
Top = 34
Width = 313
Height = 2
Align = alBottom
Shape = bsTopLine
end
< object \\ Navigator:
ı
TDBNavigator Left = 16
Top = 5
Width = 240
Height = 25
HelpContext = 6
DataSource = PartsSource1
ParentShowHint = False
ShowHint = True
TabOrder = 0
end
end
< object \\ Panel2: TPanel
Left = 0
Top = 36
Width = 313
Height = 193
Align = alTop
BevelOuter = bvNone
Develouter - Dynone

<pre>chicat \\ IIalmCamtamta.</pre>
< object \\ HelpContents:
TMenuItem
Caption = '&Contents'
OnClick = HelpBtnClick
end
< object \\ N6: TMenuItem
Caption = '-'
end
< object \\ HelpAbout:
TMenuItem
Caption = '&About'
OnClick = AboutClick
end
end
end
< object \\ PrinterSetup:
TPrinterSetupDialog
Left = 91
Top = 52
end
End
< object \\ EdPartsForm:
TEdPartsForm
Left = 274
Left = 274 Top = 90
Left = 274 Top = 90 HelpContext = 6
Left = 274 Top = 90 HelpContext = 6 ActiveControl = DBEdit2
Left = 274 Top = 90 HelpContext = 6 ActiveControl = DBEdit2 BorderIcons = [biSystemMenu,
Left = 274  Top = 90  HelpContext = 6  ActiveControl = DBEdit2  BorderIcons = [biSystemMenu, biMinimize]
Left = 274  Top = 90  HelpContext = 6  ActiveControl = DBEdit2  BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle
Left = 274  Top = 90  HelpContext = 6  ActiveControl = DBEdit2  BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle  Caption = 'Edit Parts'
Left = 274  Top = 90  HelpContext = 6  ActiveControl = DBEdit2  BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle  Caption = 'Edit Parts'  ClientHeight = 264
Left = 274  Top = 90  HelpContext = 6  ActiveControl = DBEdit2  BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle  Caption = 'Edit Parts'  ClientHeight = 264  ClientWidth = 313
Left = 274  Top = 90  HelpContext = 6  ActiveControl = DBEdit2  BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle  Caption = 'Edit Parts'  ClientHeight = 264  ClientWidth = 313  Color = clBtnFace
Left = 274  Top = 90  HelpContext = 6  ActiveControl = DBEdit2  BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle  Caption = 'Edit Parts'  ClientHeight = 264  ClientWidth = 313  Color = clBtnFace  Font.Charset =
Left = 274  Top = 90  HelpContext = 6  ActiveControl = DBEdit2  BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle  Caption = 'Edit Parts'  ClientHeight = 264  ClientWidth = 313  Color = clBtnFace  Font.Charset = DEFAULT_CHARSET
Left = 274  Top = 90  HelpContext = 6  ActiveControl = DBEdit2  BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle  Caption = 'Edit Parts'  ClientHeight = 264  ClientWidth = 313  Color = clBtnFace  Font.Charset = DEFAULT_CHARSET  Font.Color = clBlack
Left = 274  Top = 90  HelpContext = 6  ActiveControl = DBEdit2  BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle  Caption = 'Edit Parts'  ClientHeight = 264  ClientWidth = 313  Color = clBtnFace  Font.Charset =  DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11
Left = 274  Top = 90  HelpContext = 6  ActiveControl = DBEdit2  BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle  Caption = 'Edit Parts'  ClientHeight = 264  ClientWidth = 313  Color = clBtnFace  Font.Charset = DEFAULT_CHARSET  Font.Color = clBlack
Left = 274  Top = 90  HelpContext = 6  ActiveControl = DBEdit2  BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle  Caption = 'Edit Parts'  ClientHeight = 264  ClientWidth = 313  Color = clBtnFace  Font.Charset =  DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11
Left = 274  Top = 90  HelpContext = 6  ActiveControl = DBEdit2  BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle  Caption = 'Edit Parts'  ClientHeight = 264  ClientWidth = 313  Color = clBtnFace  Font.Charset = DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'
Left = 274  Top = 90  HelpContext = 6  ActiveControl = DBEdit2  BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle  Caption = 'Edit Parts'  ClientHeight = 264  ClientWidth = 313  Color = clBtnFace  Font.Charset =  DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  OldCreateOrder = True
Left = 274  Top = 90  HelpContext = 6  ActiveControl = DBEdit2  BorderIcons = [biSystemMenu, biMinimize]  BorderStyle = bsSingle  Caption = 'Edit Parts'  ClientHeight = 264  ClientWidth = 313  Color = clBtnFace  Font.Charset = DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []

Caption = 'ListPrice' end< object \\ Label6: TLabel Left = 8 Top = 122 Width = 61 Height = 13
< object \\ Label6: TLabel  Left = 8  Top = 122  Width = 61
Left = 8 Top = 122 Width = 61
Top = 122 Width = 61
Width = 61
Height = 13
Caption = 'Backordered'
end
< object \\ DBEdit2: TDBEdit
Left = 80
Top = 29
Width = 225
Height = 21
HelpContext = 6
DataField = 'Description'
DataSource = PartsSource1
TabOrder = 1
end
< object \\ DBEdit4: TDBEdit
Left = 80
Top = 74
Width = 82
Height = 21
HelpContext = 6
DataField = 'OnHand'
DataSource = PartsSource1
TabOrder = 3
end
< object \\ DBEdit5: TDBEdit
Left = 80
Top = 97
Width = 82
Height = 21
HelpContext = 6
DataField = 'OnOrder'
DataSource = PartsSource1
TabOrder = 4
end
< object \\ DBEdit7: TDBEdit
Left = 80
Top = 141
Width = 102
Height = 21
HelpContext = 6
DataField = 'Cost'
DataSource = PartsSource1
TabOrder = 6

TabOrder = 1
< object \\ Label1: TLabel
Left = 8
Top = 11
<b>Width = 33</b>
Height = 13
Caption = 'PartNo'
end
< object \\ Label2: TLabel
Left = 8
Top = 33
Width = 53
Height = 13
Caption = 'Description'
end
< object \\ Label3: TLabel
Left = 8
Top = 55
Width = 34
Height = 13
Caption = 'Vendor'
end
< object \\ Label4: TLabel
Left = 8
Top = 77
Width = 40
Height = 13
Caption = 'OnHand'
end
< object \\ Label5: TLabel
Left = 8
Top = 100
Width = 40
Height = 13
Caption = 'OnOrder'
end
< object \\ Label7: TLabel
Left = 8
Top = 144
<b>Width = 21</b>
Height = 13
Caption = 'Cost'
end
< object \\ Label8: TLabel
Left = 8
Top = 166
Width = 40
Height = 13
II cigiti — 15

Left = 147
Top = 235
Width = 75
Height = 25
Caption = 'OK'
Default = True
ModalResult = 1
TabOrder = 2
end
< object \\ CancelButton: TButton
Left = 232
Top = 235
Width = 75
Height = 25
Cancel = True
Caption = 'Cancel'
ModalResult = 2
TabOrder = 3
end
< object \\ PartsSource1:
<b>TDataSource</b>
DataSet = MastData.Parts
Left = 24
Top = 232
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
< object \\ EdCustForm:
TEdCustForm
Left = 253
Top = 134
•
HelpContext = 1
ActiveControl = DBEdName
BorderIcons = [biSystemMenu,
biMinimize]
BorderStyle = bsDialog
Caption = 'Edit Customers'
ClientHeight = 307
ClientWidth = 376
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'

,
end
< object \\ DBEdit8: TDBEdit
Left = 80
Top = 163
Width = 102
Height = 21
HelpContext = 6
DataField = 'ListPrice'
DataSource = PartsSource1
TabOrder = 7
end
< object \\ DBEdPartNo:
TDBEdit
Left = 80
Top = 6
Width = 102
Height = 21
HelpContext = 6
DataField = 'PartNo'
DataSource = PartsSource1
TabOrder = 0
end
< object \\ DBEdit3: TDBEdit
Left = 80
Top = 119
Width = 45
Height = 21
HelpContext = 6
DataField = 'BackOrd'
DataSource = PartsSource1
TabOrder = 5
end
< object \\ DataComboBox1:
TDBLookupComboBox
Left = 80
Top = 51
Width = 225
Height = 21
DataField = 'VendorNo'
DataSource = PartsSource1
KeyField = 'VendorNo'
ListField = 'VendorName'
ListSource =
MastData.VendorSource
TabOrder = 2
end
V == Vr
end
< object \\ OKButton: TButton

ou d
end TI I I TI I I
< object \\ Label6: TLabel
Left = 124
Top = 134
Width = 25
Height = 13
Caption = 'State'
end
< object \\ Label7: TLabel
Left = 192
Top = 134
Width = 43
Height = 13
Caption = 'Zip Code'
end
< object \\ Label8: TLabel
Left = 289
Top = 132
Width = 36
Height = 13
Caption = 'Country'
end
< object \\ Label9: TLabel
Left = 248
Top = 53
Width = 31
Height = 13
Caption = 'Phone'
end
< object \\ Label11: TLabel
Left = 284
Top = 186
Width = 44
Width = 44 Height = 13
Caption = 'Tax Rate'
end
< object \\ Label13: TLabel
Left = 144
Top = 188
Width = 58
Height = 13
Caption = 'Last Invoice'
end
< object \\ Label14: TLabel
Left = 248
Top = 92
Width = 17
Height = 13

Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
OnCloseQuery = FormCloseQuery
PixelsPerInch = 96
TextHeight = 13
< object \\ Panel2: TPanel
Left = 0
Top = 39
Width = 376
Height = 228
Align = alClient
BevelOuter = bvNone
TabOrder = 0
< object \\ Label1: TLabel
Left = 248
Top = 12
Width = 35
Height = 13
Caption = 'CustNo'
end
< object \\ Label2: TLabel
Left = 20
Top = 12
Width = 44
Height = 13
Caption = 'Company'
end
< object \\ Label3: TLabel
Left = 20
Top = 53
Width = 25
Height = 13
Caption = 'Add1'
end
< object \\ Label4: TLabel
Left = 20
Top = 93
Width = 25
Height = 13
Caption = 'Add2'
end
< object \\ Label5: TLabel
Left = 20
Top = 133
Width = 17
Height = 13
Caption = 'City'

M
MastData.CustSource
TabOrder = 1
end
< object \\ DBEdit4: TDBEdit
Left = 19
Top = 108
Width = 205
Height = 21
DataField = 'Addr2'
DataSource =
MastData.CustSource
TabOrder = 2
end
< object \\ DBEdit5: TDBEdit
Left = 19
Top = 148
Width = 98
Height = 21
DataField = 'City'
DataSource =
MastData.CustSource
TabOrder = 3
end
< object \\ DBEdit6: TDBEdit
Left = 123
Top = 148
Width = 62
Height = 21
DataField = 'State'
DataSource =
MastData.CustSource
TabOrder = 4
end
< object \\ DBEdit7: TDBEdit
Left = 191
Top = 148
Width = 90
Height = 21
DataField = 'Zip'
DataFicid - Zip DataSource =
MastData.CustSource
TabOrder = 5
end
< object \\ DBEdit8: TDBEdit
Left = 288
Top = 148
Width = 66
Height = 21

Caption = 'Fax'
end
< object \\ Bevel2: TBevel
Left = 8
Top = 178
Width = 345
Height = 5
Shape = bsTopLine
end
< object \\ Label10: TLabel
Left = 16
Top = 188
-
Width = 37
Height = 13
Caption = 'Contact'
end
< object \\ DBEdCustNo:
TDBEdit
Left = 247
Top = 28
<b>Width = 69</b>
Height = 21
Color = clSilver
DataField = 'CustNo'
DataSource =
MastData.CustSource
Enabled = False
ReadOnly = True
TabOrder = 12
end
< object \\ DBEdName:
TDBEdit
Left = 19
Top = 28
Width = 205
Height = 21
DataField = 'Company'
DataSource =
MastData.CustSource
TabOrder = 0
end
< object \\ DBEdit3: TDBEdit
Left = 19
Top = 68
Width = 205
Height = 21
DataField = 'Addr1'
DataSource =
Datasourte -

Left = 247
Top = 106
Width = 106
Height = 21
DataField = 'FAX'
DataSource =
MastData.CustSource
TabOrder = 8
end
end
< object \\ Panel: TPanel
Left = 0
Top = 0
Width = 376
Height = 39
Align = alTop
BevelOuter = bvNone
BorderWidth = 2
TabOrder = 1
< object \\ PrintBtn: TSpeedButton
Left = 339
Top = 5
Width = 25
Height = 25
Hint = 'Print form image'
Glyph.Data = {
333333333333300000}
OnClick = PrintBtnClick
end
< object \\ Bevel1: TBevel
Left = 2
Top = 35
Width = 372
Height = 2
Align = alBottom
Shape = bsTopLine
end
< object \\ DBNavigator:
TDBNavigator
Left = 19
<b>Top</b> = <b>5</b>
Width = 250
Height = 25
DataSource =
MastData.CustSource
ParentShowHint = False
ShowHint = True
MICHIEL IIUC

<u> </u>
DataField = 'Country'
DataSource =
MastData.CustSource
TabOrder = 6
end
< object \\ DBEdit9: TDBEdit
Left = 247
Top = 68
Width = 106
Height = 21
DataField = 'Phone'
DataField - Filone  DataSource =
MastData.CustSource
TabOrder = 7
end
< object \\ DBEdit11:
TDBEdit
Left = 283
Top = 203
Width = 54
Height = 21
DataField = 'TaxRate'
DataSource =
MastData.CustSource
TabOrder = 11
end
< object \\ DBEdit12:
TDBEdit
Left = 15
Top = 203
Width = 118
Height = 21
DataField = 'Contact'
DataField - Contact  DataSource =
MastData.CustSource
TabOrder = 9
end (I) DDE H (EDDE H)
< object \\ DBEdInv: TDBEdit
Left = 143
Top = 203
Width = 131
Height = 21
DataField = 'LastInvoiceDate'
DataSource =
MastData.CustSource
TabOrder = 10
end
< object \\ DBEdit14:
TDBEdit

Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
OnShow = FormShow
PixelsPerInch = 96
TextHeight = 13
< object \\ Panel1: TPanel
Left = 0
Top = 0
Width = 384
Height = 37
HelpContext = 4
Align = alTop
BevelOuter = bvNone
BorderWidth = 3
TabOrder = 0
< object \\ ActivateBtn:
TSpeedButton
Left = 284
Top = 4
Width = 85
Height = 25
AllowAllUp = True
GroupIndex = 1
Caption = 'Backorders'
OnClick = ActivateQuery
end
< object \\ Bevel1: TBevel
Left = 3
Top = 32
Width = 378
Height = 2
Align = alBottom
Shape = bsTopLine
end
< object \\ Navigator:
TDBNavigator
Left = 8
Top = 4
Width = 135
Height = 25
DataSource =
MastData.PartsSource
masidata.1 altisoulitt

T. I.O. I
TabOrder = 0
end
end
< object \\ Panel1: TPanel
Left = 0
Top = 267
Width = 376
Height = 40
Align = alBottom
BevelOuter = bvNone
TabOrder = 2
< object \\ CancelButton:
TButton
Left = 292
Top = 8
Width = 75
Height = 25
Cancel = True
Caption = 'Cancel'
ModalResult = 2
TabOrder = 0
end
< object \\ OKButton:
TButton
Left = 204
Top = 8
<b>Width</b> = 75
Height = 25
Caption = 'OK'
Default = True
ModalResult = 1
TabOrder = 1
end
end
End
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
< object \\ BrPartsForm:
TBrPartsForm
Left = 229
Top = 151
HelpContext = 4
BorderIcons = [biSystemMenu]
BorderStyle = bsDialog
Caption = 'Browse Parts'
ClientHeight = 235
ClientWidth = 384
Color = clBtnFace

TabOrder = 2
TitleFont.Charset =
DEFAULT_CHARSET
TitleFont.Color = clBlack
TitleFont.Height = -11
TitleFont.Name = 'MS Sans Serif'
TitleFont.Style = []
OnDblClick = EditBtnClick
Columns = <
item
Expanded = False
FieldName = 'PartNo'
Visible = True
end
item
Expanded = False
FieldName = 'Description'
Visible = True
visible = 1 rue end
item
Expanded = False
FieldName = 'OnHand'
Visible = True
end
item
Expanded = False
FieldName = 'OnOrder'
Visible = True
end
item
Expanded = False
FieldName = 'BackOrd'
Visible = True
end>
end
End
111111111111111111111111111111111111111
< object \\ BrDateForm:
TBrDateForm
Left = 221
Top = 143
BorderStyle = bsDialog
Caption = 'Select a Date'
ClientHeight = 220
ClientWidth = 362

VisibleButtons = [nbFirst,
nbPrior, nbNext, nbLast, nbRefresh]
ParentShowHint = False
ShowHint = True
TabOrder = 0
end
< object \\ EditBtn: TButton
Left = 177
Top = 4
Width = 74
Height = 25
Caption = '&Edit'
ModalResult = 1
TabOrder = 1
OnClick = EditBtnClick
end
end
< object \\ Panel3: TPanel
Left = 0
Top = 196
Width = 384
Height = 39
Align = alBottom
BevelOuter = bvNone
TabOrder = 1
< object \\ CloseBtn: TButton
Left = 300
<b>Top = 9</b>
Width = 74
Height = 25
Cancel = True
Caption = '&Close'
Default = True
ModalResult = 2
TabOrder = 0
OnClick = CloseBtnClick
end
end
< object \\ PartsGrid: TDBGrid
Left = 0
Top = 37
Width = 384
Height = 159
Align = alClient
DataSource =
MastData.PartsSource
Options = [dgTitles, dgColLines,
dgRowLines, dgRowSelect
g / g ······

Height = 20
Glyph.Data = {
33003333333333333300}
OnClick = NextMonthBtnClick
end
< object \\ Calendar1:
TCalendar
Left = 22
Top = 41
Width = 318
Height = 136
Color = clInfoBk
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
StartOfWeek = 0
TabOrder = 0
UseCurrentDate = False
OnChange = Calendar1Change
end
< object \\ OkBtn: TButton
Left = 204
Top = 189
Width = 73
Height = 25
Caption = '&OK'
Default = True
ModalResult = 1
TabOrder = 1
end
< object \\ CancelBtn: TButton
Left = 284
Top = 189
Width = 73
Height = 25
Caption = '&Cancel'
ModalResult = 2
TabOrder = 2
end
End
\\\\\

Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
PixelsPerInch = 96
TextHeight = 13
< object \\ Bevel1: TBevel
Left = 16
Top = 8
Width = 330
Height = 177
Shape = bsFrame
end
< object \\ TitleLabel: TLabel
Left = 54
Top = 18
Width = 257
Height = 13
Alignment = taCenter
AutoSize = False
Caption = 'February, 1995'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
< object \\ PrevMonthBtn:
TSpeedButton
Left = 22
Top = 16
Width = 20
Height = 20
Glyph.Data = { OnClick = PrevMonthBtnClick
end
< object \\ NextMonthBtn:
TSpeedButton
Left = 320
Top = 16
Width = 20

<del>-</del>
Font.Style = [fsBold]
ParentFont = False
IsControl = True
end
< object \\ Version: TLabel
Left = 128
Top = 40
Width = 65
Height = 13
Caption = 'Version 3.0'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
IsControl = True
end
< object \\ Copyright: TLabel
Left = 16
Top = 136
Width = 258
Height = 13
Caption = 'Copyright 1995,1997
Marine Adventures, Inc.'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
IsControl = True
end
end
< object \\ Button1: TButton
Left = 114
Top = 180
Width = 75
Height = 25
Cancel = True
Caption = '&OK'
Default = True
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11

< object \\ AboutBox:
TAboutBox
Left = 239
Top = 127
BorderStyle = bsDialog
Caption = 'About'
ClientHeight = 210
ClientWidth = 304
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clWindowText
Font.Height = -13
Font.Name = 'System'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
PixelsPerInch = 96
TextHeight = 16
< object \\ Panel1: TPanel
Left = 8
Top = 8
Width = 289
Height = 161
BevelInner = byRaised
BevelOuter = bvLowered
TabOrder = 0
< object \\ ProgramIcon:
TImage
Left = 8
Top = 8
Width = 105
Height = 105
Picture.Data = {
IsControl = True
end
< object \\ ProductName:
TLabel
Left = 128
Top = 16
Width = 35
Height = 13
Caption = 'MAST'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'

end
< object \\ OKBtn: TButton
Left = 2
Top = 160
<b>Width</b> = 75
Height = 25
Caption = '&OK'
Default = True
ModalResult = 1
TabOrder = 1
end
< object \\ DBLookupListBox1:
TDBLookupListBox
Left = 5
<b>Top = 16</b>
<b>Width</b> = 153
Height = 134
KeyField = 'OrderNo'
ListField = 'OrderNo'
ListSource =
MastData.OrdersSource
TabOrder = 2
end
End
انتهى الملف النصي ع ل
\\\\\\

Font.Name = 'MS Sans Serif'
Font.Style = []
ModalResult = 1
ParentFont = False
TabOrder = 1
end
End
\\\\\\
< object \\ PickOrderNoDlg:
TPickOrderNoDlg
Left = 236
Top = 124 AutoScroll = False
Caption = 'Order No.'
ClientHeight = 190 ClientWidth = 166
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
OnShow = FormShow
PixelsPerInch = 96
TextHeight = 13
< object \\ Label1: TLabel
Left = 5
Top = 1
Width = 94
Height = 13
Caption = '&Select an Order No.'
FocusControl =
DBLookupListBox1
end
< object \\ CancelBtn: TButton
Left = 86
Top = 160
<b>Width</b> = 75
Height = 25
Cancel = True
Caption = '&Cancel'
ModalResult = 2
TabOrder = 0

unit SrchDlg;	Procedures 43
interface	الملف التنفيذي لبرنامج
uses SysUtils, Windows, Messages,	(MastApp)
Classes, Graphics, Controls, Forms, Dialogs, DBTables, DB,	إعداد
StdCtrls, ExtCtrls, Grids, DBGrids, Buttons;	علاء الدين اللباد
,	هاتف ۷۱۳۹۷۹۹۹۰
type TSearchDlg = class(TForm)	unit Splash;
DataSource: TDataSource; DBGrid1: TDBGrid;	interface
OKBtn: TButton; CancelBtn: TButton;	uses
SearchEd: TEdit;	SysUtils, Windows, Messages, Classes, Graphics, Controls,
OrderCombo: TComboBox; Label1: TLabel;	Forms, Dialogs, StdCtrls, ExtCtrls;
Label2: TLabel; SearchButton: TSpeedButton;	type
<pre>&lt;&gt; procedure DBGrid1DblClick(Sender:</pre>	TSplashForm = class(TForm) Panel1: TPanel;
TObject);	Label3: TLabel; Bevel1: TBevel;
<> procedure SearchButtonClick(Sender:	Label1: TLabel; Image1: TImage;
TObject); <> procedure	<> procedure
OrderComboChange(Sender: TObject);	Panel1Click(Sender: TObject); end;
<> procedure SearchEdKeyPress(Sender:	var
TObject; var Key: Char);	SplashForm: TSplashForm;
SearchEdChange(Sender: TObject);	implementation
<pre>&lt;&gt; procedure FormCreate(Sender: TObject);</pre>	{\$R *.dfm}
private SrchFld: TField;	procedure
function GetCustNo: Double; <> procedure	TSplashForm.Panel1Click(Sender: TObject);
SetCustNo(NewCustNo: Double);	begin
function GetPartNo: Double; <> procedure	end;
SetPartNo(NewPartNo: Double); public	end.

function	property PartNo: Double read
TSearchDlg.ShowModalCust:	GetPartNo write SetPartNo;
Integer;	property CustNo: Double read
begin	GetCustNo write SetCustNo;
OrderCombo.Items.Clear;	function ShowModalCust:
order compositems, crear,	Integer;
OrderCombo.Items.Add('Company'	function ShowModalParts:
);	Integer;
OrderCombo.Items.Add('CustNo');	end;
OrderCombo.ItemIndex := 0;	Citu,
Datasource.Dataset :=	viow.
	Var
MastData.Cust;	SearchDlg: TSearchDlg;
OrderComboChange(nil);	
Caption := 'Select a Customer';	implementation
Result := ShowModal;	
end;	uses DataMod;
function	{\$R *.dfm}
TSearchDlg.ShowModalParts:	
Integer;	function TSearchDlg.GetCustNo:
begin	Double;
OrderCombo.Items.Clear;	begin
,	Result :=
OrderCombo.Items.Add('Descriptio	MastData.CustCustNo.Value;
n');	end;
OrderCombo.Items.Add('PartNo');	onu,
OrderCombo.ItemIndex := 0;	procedure
Datasource.Dataset :=	TSearchDlg.SetCustNo(NewCustNo:
MastData.Parts;	Double);
OrderComboChange(nil);	begin
Caption := 'Select a Part';	MastData.Cust.Locate('CustNo',
	· ·
Result := ShowModal;	NewCustNo, []);
end;	end;
nuocoduno	function TCoonst DL- C-4D4N
procedure TSooreh Dlg DPC rid 1 DblClick (Sond	function TSearchDlg.GetPartNo:
TSearchDlg.DBGrid1DblClick(Send	Double;
er: TObject);	begin
begin	Result :=
ModalResult := mrOK;	MastData.PartsPartNo.Value;
end;	end;
procedure	procedure
TSearchDlg.SearchButtonClick(Sen	TSearchDlg.SetPartNo(NewPartNo:
der: TObject);	Double);
begin	begin
if not	MastData.Parts.Locate('PartNo',
Datasource.Dataset.Locate(OrderCo	NewPartNo, []);
mbo.Text, SearchEd.Text,	end;
[loCaseInsensitive, loPartialKey])	,
• • • • • • • • • • • • • • • • • • • •	

	then
interface	MessageDlg('No matching record
Interface	found.', mtInformation, [mbOK], 0);
uses	end;
SysUtils, Windows, Messages,	onu,
Classes, Graphics, Controls,	procedure
Forms, Dialogs, DBTables, DB,	TSearchDlg.OrderComboChange(Se
StdCtrls, ExtCtrls, Grids, DBGrids,	nder: TObject);
Buttons;	begin
Duttons,	SrchFld :=
type	Datasource.Dataset.FieldByName(O
TSearchDlg = class(TForm)	rderCombo.Text);
DataSource: TDataSource;	SearchEd.Text := ";
DBGrid1: TDBGrid;	end;
OKBtn: TButton;	,
CancelBtn: TButton;	procedure
SearchEd: TEdit;	TSearchDlg.SearchEdKeyPress(Sen
OrderCombo: TComboBox;	der: TObject; var Key: Char);
Label1: TLabel;	begin
Label2: TLabel;	if Assigned(SrchFld) and (Key > ' ')
SearchButton: TSpeedButton;	and not (SrchFld.IsValidChar(Key))
<> procedure	then
DBGrid1DblClick(Sender:	begin
TObject);	MessageBeep(0);
<> procedure	Key := #0;
SearchButtonClick(Sender:	end;
TObject);	end;
<> procedure	,
OrderComboChange(Sender:	procedure
TObject);	TSearchDlg.SearchEdChange(Sende
<> procedure	r: TObject);
SearchEdKeyPress(Sender:	begin
TObject; var Key: Char);	SearchButton.Enabled :=
<> procedure	SearchEd.Text <> '';
SearchEdChange(Sender: TObject);	end;
<> procedure	
FormCreate(Sender: TObject);	procedure
private	TSearchDlg.FormCreate(Sender:
SrchFld: TField;	TObject);
function GetCustNo: Double;	begin
<> procedure	
SetCustNo(NewCustNo: Double);	end;
function GetPartNo: Double;	
<> procedure	end.
SetPartNo(NewPartNo: Double);	
public	•
property PartNo: Double read	
GetPartNo write SetPartNo;	
property CustNo: Double read	unit SrchDlg;

hagin	CatCustNo wwite SatCustNo.
begin	GetCustNo write SetCustNo;
OrderCombo.Items.Clear;	function ShowModalCust:
	Integer;
OrderCombo.Items.Add('Company'	function ShowModalParts:
);	Integer;
OrderCombo.Items.Add('CustNo');	end;
OrderCombo.ItemIndex := 0;	
Datasource.Dataset :=	var
MastData.Cust;	SearchDlg: TSearchDlg;
OrderComboChange(nil);	
Caption := 'Select a Customer';	implementation
Result := ShowModal;	
end;	uses DataMod;
function	{\$R *.dfm}
TSearchDlg.ShowModalParts:	
Integer;	function TSearchDlg.GetCustNo:
begin	Double;
OrderCombo.Items.Clear;	begin
Order compositems. cicar;	Result :=
OrderCombo.Items.Add('Descriptio	
n');	MastData.CustCustNo.Value;
- / /	end;
OrderCombo.Items.Add('PartNo');	
OrderCombo.ItemIndex := 0;	procedure
Datasource.Dataset :=	TSearchDlg.SetCustNo(NewCustNo:
MastData.Parts;	Double);
OrderComboChange(nil);	begin
Caption := 'Select a Part';	MastData.Cust.Locate('CustNo',
Result := ShowModal;	NewCustNo, []);
end;	end;
procedure	function TSearchDlg.GetPartNo:
TSearchDlg.DBGrid1DblClick(Send	Double;
er: TObject);	begin
begin	Result :=
ModalResult := mrOK;	MastData.PartsPartNo.Value;
end;	end;
procedure	procedure
TSearchDlg.SearchButtonClick(Sen	TSearchDlg.SetPartNo(NewPartNo:
der: TObject);	Double);
begin	begin
if not	MastData.Parts.Locate('PartNo',
Datasource.Dataset.Locate(OrderCo	NewPartNo, []);
mbo.Text, SearchEd.Text,	end;
[loCaseInsensitive, loPartialKey])	varus,
then	function
MessageDlg('No matching record	TSearchDlg.ShowModalCust:
found.', mtInformation, [mbOK], 0);	
round, marrior mation, [mbOK], v);	Integer;

Classes, Graphics, Controls,	end;
	enu,
Forms, Dialogs, DBTables, DB,	
StdCtrls, ExtCtrls, Grids, DBGrids,	procedure
Buttons;	TSearchDlg.OrderComboChange(Se
	nder: TObject);
type	begin
TSearchDlg = class(TForm)	SrchFld :=
DataSource: TDataSource;	Datasource.Dataset.FieldByName(O
DBGrid1: TDBGrid;	rderCombo.Text);
OKBtn: TButton;	SearchEd.Text := ";
CancelBtn: TButton;	end;
SearchEd: TEdit;	
OrderCombo: TComboBox;	procedure
Label1: TLabel;	TSearchDlg.SearchEdKeyPress(Sen
Label2: TLabel;	der: TObject; var Key: Char);
SearchButton: TSpeedButton;	begin
<> procedure	if Assigned(SrchFld) and (Key > ' ')
DBGrid1DblClick(Sender:	and not (SrchFld.IsValidChar(Key))
TObject);	then
<> procedure	begin
SearchButtonClick(Sender:	MessageBeep(0);
TObject);	Key := #0;
<> procedure	end;
OrderComboChange(Sender:	end;
TObject);	cnu,
<> procedure	nyoooduyo
SearchEdKeyPress(Sender:	procedure TSoowhDlg SoowhEdChange(Sondo
· `	TSearchDlg.SearchEdChange(Sende r: TObject);
TObject; var Key: Char);	<b>"</b> //
<> procedure	begin
SearchEdChange(Sender: TObject);	SearchButton.Enabled :=
<> procedure	SearchEd.Text <> '';
FormCreate(Sender: TObject);	end;
private	
SrchFld: TField;	procedure
function GetCustNo: Double;	TSearchDlg.FormCreate(Sender:
<> procedure	TObject);
SetCustNo(NewCustNo: Double);	begin
function GetPartNo: Double;	
<> procedure	end;
SetPartNo(NewPartNo: Double);	
public	end.
property PartNo: Double read	
GetPartNo write SetPartNo;	
property CustNo: Double read	unit SrchDlg;
GetCustNo write SetCustNo;	<b>3</b> /
function ShowModalCust:	interface
Integer;	
function ShowModalParts:	uses
Integer;	SysUtils, Windows, Messages,
	sysums, windows, Messages,

	_
OrderCombo.Items.Add('CustNo');	end;
OrderCombo.ItemIndex := 0;	
Datasource.Dataset :=	var
MastData.Cust;	SearchDlg: TSearchDlg;
OrderComboChange(nil);	
Caption := 'Select a Customer';	implementation
Result := ShowModal;	
end;	uses DataMod;
onu,	uses Dutaiviou,
function	{\$R *.dfm}
	(\$K .uiiii)
TSearchDlg.ShowModalParts:	e d' TEC LIDI C (C A)
Integer;	function TSearchDlg.GetCustNo:
begin	Double;
OrderCombo.Items.Clear;	begin
	Result :=
OrderCombo.Items.Add('Descriptio	MastData.CustCustNo.Value;
n');	end;
OrderCombo.Items.Add('PartNo');	
OrderCombo.ItemIndex := 0;	procedure
Datasource.Dataset :=	TSearchDlg.SetCustNo(NewCustNo:
MastData.Parts;	Double);
OrderComboChange(nil);	begin
Caption := 'Select a Part';	MastData.Cust.Locate('CustNo',
Result := ShowModal;	NewCustNo, []);
end;	end;
enu,	enu,
nuonoduno	function TS and Dla CotPartNo.
procedure	function TSearchDlg.GetPartNo:
TSearchDlg.DBGrid1DblClick(Send	Double;
er: TObject);	begin
begin	Result :=
ModalResult := mrOK;	MastData.PartsPartNo.Value;
end;	end;
procedure	nwoooduwo
procedure	procedure
TSearchDlg.SearchButtonClick(Sen	TSearchDlg.SetPartNo(NewPartNo:
-	-
TSearchDlg.SearchButtonClick(Sen	TSearchDlg.SetPartNo(NewPartNo: Double);
TSearchDlg.SearchButtonClick(Sen der: TObject);	TSearchDlg.SetPartNo(NewPartNo: Double); begin
TSearchDlg.SearchButtonClick(Sen der: TObject); begin if not	TSearchDlg.SetPartNo(NewPartNo: Double); begin MastData.Parts.Locate('PartNo',
TSearchDlg.SearchButtonClick(Sen der: TObject); begin if not Datasource.Dataset.Locate(OrderCo	TSearchDlg.SetPartNo(NewPartNo: Double); begin MastData.Parts.Locate('PartNo', NewPartNo, []);
TSearchDlg.SearchButtonClick(Sen der: TObject); begin if not Datasource.Dataset.Locate(OrderCo mbo.Text, SearchEd.Text,	TSearchDlg.SetPartNo(NewPartNo: Double); begin MastData.Parts.Locate('PartNo',
TSearchDlg.SearchButtonClick(Sen der: TObject); begin if not Datasource.Dataset.Locate(OrderCombo.Text, SearchEd.Text, [loCaseInsensitive, loPartialKey])	TSearchDlg.SetPartNo(NewPartNo: Double); begin MastData.Parts.Locate('PartNo', NewPartNo, []); end;
TSearchDlg.SearchButtonClick(Sen der: TObject); begin if not Datasource.Dataset.Locate(OrderCombo.Text, SearchEd.Text, [loCaseInsensitive, loPartialKey]) then	TSearchDlg.SetPartNo(NewPartNo: Double); begin MastData.Parts.Locate('PartNo', NewPartNo, []); end; function
TSearchDlg.SearchButtonClick(Sen der: TObject); begin if not Datasource.Dataset.Locate(OrderCombo.Text, SearchEd.Text, [loCaseInsensitive, loPartialKey]) then MessageDlg('No matching record	TSearchDlg.SetPartNo(NewPartNo: Double); begin MastData.Parts.Locate('PartNo', NewPartNo, []); end; function TSearchDlg.ShowModalCust:
TSearchDlg.SearchButtonClick(Sen der: TObject); begin if not Datasource.Dataset.Locate(OrderCombo.Text, SearchEd.Text, [loCaseInsensitive, loPartialKey]) then MessageDlg('No matching record found.', mtInformation, [mbOK], 0);	TSearchDlg.SetPartNo(NewPartNo: Double); begin MastData.Parts.Locate('PartNo', NewPartNo, []); end; function TSearchDlg.ShowModalCust: Integer;
TSearchDlg.SearchButtonClick(Sen der: TObject); begin if not Datasource.Dataset.Locate(OrderCombo.Text, SearchEd.Text, [loCaseInsensitive, loPartialKey]) then MessageDlg('No matching record	TSearchDlg.SetPartNo(NewPartNo: Double); begin MastData.Parts.Locate('PartNo', NewPartNo, []); end; function TSearchDlg.ShowModalCust: Integer; begin
TSearchDlg.SearchButtonClick(Sen der: TObject); begin if not Datasource.Dataset.Locate(OrderCombo.Text, SearchEd.Text, [loCaseInsensitive, loPartialKey]) then MessageDlg('No matching record found.', mtInformation, [mbOK], 0); end;	TSearchDlg.SetPartNo(NewPartNo: Double); begin MastData.Parts.Locate('PartNo', NewPartNo, []); end; function TSearchDlg.ShowModalCust: Integer;
TSearchDlg.SearchButtonClick(Sen der: TObject); begin if not Datasource.Dataset.Locate(OrderCombo.Text, SearchEd.Text, [loCaseInsensitive, loPartialKey]) then MessageDlg('No matching record found.', mtInformation, [mbOK], 0); end;  procedure	TSearchDlg.SetPartNo(NewPartNo: Double); begin MastData.Parts.Locate('PartNo', NewPartNo, []); end; function TSearchDlg.ShowModalCust: Integer; begin OrderCombo.Items.Clear;
TSearchDlg.SearchButtonClick(Sen der: TObject); begin if not Datasource.Dataset.Locate(OrderCombo.Text, SearchEd.Text, [loCaseInsensitive, loPartialKey]) then MessageDlg('No matching record found.', mtInformation, [mbOK], 0); end;	TSearchDlg.SetPartNo(NewPartNo: Double); begin MastData.Parts.Locate('PartNo', NewPartNo, []); end; function TSearchDlg.ShowModalCust: Integer; begin

Γ,	, .
type	begin
TQueryCustDlg = class(TForm)	SrchFld :=
Label1: TLabel;	Datasource.Dataset.FieldByName(O
Label2: TLabel;	rderCombo.Text);
FromEdit: TEdit;	SearchEd.Text := ";
ToEdit: TEdit;	end;
CancelBtn: TButton;	
OkBtn: TButton;	procedure
Msglab: TLabel;	TSearchDlg.SearchEdKeyPress(Sen
PopupCalBtnFrom:	der: TObject; var Key: Char);
TSpeedButton;	begin
PopupCalToBtn: TSpeedButton;	if Assigned(SrchFld) and (Key > ' ')
Bevel1: TBevel;	and not (SrchFld.IsValidChar(Key))
<> procedure	then
OkBtnClick(Sender: TObject);	begin
<> procedure	MessageBeep(0);
PopupCalBtnFromClick(Sender:	Key := #0;
TObject);	end;
<> procedure	end;
PopupCalToBtnClick(Sender:	,
TObject);	procedure
<> procedure	TSearchDlg.SearchEdChange(Sende
FormCreate(Sender: TObject);	r: TObject);
private	begin
function GetFromDate:	SearchButton.Enabled :=
TDateTime;	SearchEd.Text <> ";
function GetToDate: TDateTime;	end;
<> procedure	chu,
SetFromDate(NewDate:	procedure
TDateTime);	TSearchDlg.FormCreate(Sender:
<> procedure	TObject);
SetToDate(NewDate: TDateTime);	begin
public	begin
property FromDate: TDateTime	ands
read GetFromDate write	end;
	and
SetFromDate;	end.
property ToDate: TDateTime	
read GetToDate write SetToDate;	
end;	1.0.0
	unit QryCust;
var	
QueryCustDlg: TQueryCustDlg;	interface
implementation	uses
	SysUtils, Windows, Messages,
{\$R *.dfm}	Classes, Graphics, Controls,
	Forms, Dialogs, StdCtrls, ExtCtrls,
uses Pickdate;	Buttons;

ChamManage (Lincolt Library	
ShowMessage(' Invalid date	procedure
specified');	TQueryCustDlg.SetFromDate(NewD
ModalResult := mrNone;	ate: TDateTime);
end;	begin
end;	FromEdit.Text :=
	DateToStr(NewDate);
procedure	end;
TQueryCustDlg.PopupCalBtnFrom	
Click(Sender: TObject);	procedure
begin	TQueryCustDlg.SetToDate(NewDat
BrDateForm.Date :=	e: TDateTime);
StrToDate(FromEdit.Text); { start	begin
with current date }	ToEdit.Text :=
if BrDateForm.ShowModal =	DateToStr(NewDate);
mrOk then	end;
FromEdit.Text :=	· · · · · · · · · · · · · · · · · · ·
DateToStr(BrDateForm.Date);	function
end;	TQueryCustDlg.GetFromDate:
Ciru,	TDateTime;
	,
procedure	begin
TQueryCustDlg.PopupCalToBtnCli	if FromEdit.Text = " then Result :=
ck(Sender: TObject);	0
begin	else Result :=
BrDateForm.Date :=	StrToDate(FromEdit.Text);
StrToDate(ToEdit.Text); { start	end;
with current date }	
if BrDateForm.ShowModal =	function
mrOk then	TQueryCustDlg.GetToDate:
ToEdit.Text :=	TDateTime;
DateToStr(BrDateForm.Date);	begin
end;	if ToEdit.Text = " then Result := 0
	else Result :=
procedure	StrToDate(ToEdit.Text);
TQueryCustDlg.FormCreate(Sender	end;
: TObject);	,
begin	procedure
MsgLab.Caption := 'Customers	TQueryCustDlg.OkBtnClick(Sender
with LastInvoiceDate ranging:';	: TObject);
FromDate := EncodeDate(1995, 01,	begin
01);	try
ToDate := Now;	if (ToDate <> 0) and (ToDate <
end;	FromDate) then
ciru,	,
and	begin ShowMessage(!!!TO!! date
end.	ShowMessage("TO" date
	cannot be less than "FROM" date');
	ModalResult := mrNone;
	end
unit Pickrep;	else ModalResult := mrOk;
	except

	interface
end.	interrace
Citu.	uses Windows, Classes, Graphics,
	Forms, Controls, Buttons,
	StdCtrls, ExtCtrls;
'4 P' 1	Stuctris, Exictris;
unit Pickrep;	
	type
interface	TPickRpt = class(TForm)
	OKBtn: TButton;
uses Windows, Classes, Graphics,	CloseBtn: TButton;
Forms, Controls, Buttons,	ViewBtn: TButton;
StdCtrls, ExtCtrls;	ReportType: TRadioGroup;
	<> procedure
type	OKBtnClick(Sender: TObject);
TPickRpt = class(TForm)	<> procedure
OKBtn: TButton;	ViewBtnClick(Sender: TObject);
CloseBtn: TButton;	<> procedure
ViewBtn: TButton;	FormCreate(Sender: TObject);
ReportType: TRadioGroup;	public
	Preview : Boolean;
<> procedure	·
OKBtnClick(Sender: TObject);	end;
<> procedure	
ViewBtnClick(Sender: TObject);	var
<> procedure	PickRpt: TPickRpt;
FormCreate(Sender: TObject);	
public	implementation
Preview : Boolean;	
end;	<b>{\$R *.dfm}</b>
var	procedure
PickRpt: TPickRpt;	TPickRpt.OKBtnClick(Sender:
• '	TObject);
implementation	begin
	Preview := False;
{\$R *.dfm}	end;
(wit mini)	
procedure	procedure
TPickRpt.OKBtnClick(Sender:	TPickRpt.ViewBtnClick(Sender:
TObject);	TObject);
9 //	• //
begin	begin
Preview := False;	Preview := True;
end;	end;
procedure	procedure
TPickRpt.ViewBtnClick(Sender:	TPickRpt.FormCreate(Sender:
TObject);	TObject);
begin	begin
Preview := True;	
end;	end;
/	

procedure	procedure
TPickRpt.ViewBtnClick(Sender:	TPickRpt.FormCreate(Sender:
TObject);	TObject);
begin	begin
Preview := True;	
end;	end;
procedure	end.
TPickRpt.FormCreate(Sender:	
TObject);	
begin	
	unit Pickrep;
end;	
	interface
end.	
	uses Windows, Classes, Graphics,
	Forms, Controls, Buttons,
unit PickInve;	StdCtrls, ExtCtrls;
	Source State States
interface	type
	TPickRpt = class(TForm)
uses	OKBtn: TButton;
Windows, Messages, SysUtils,	CloseBtn: TButton;
Classes, Graphics, Controls, Forms,	ViewBtn: TButton;
Dialogs,	ReportType: TRadioGroup;
StdCtrls, DBCtrls;	<> procedure
Stuctus, DBCtris,	OKBtnClick(Sender: TObject);
type	<> procedure
TPickOrderNoDlg = class(TForm)	ViewBtnClick(Sender: TObject);
CancelBtn: TButton;	-
ŕ	<pre>&lt;&gt; procedure FormCreate(Sender: TObject);</pre>
OKBtn: TButton;	
DBLookupListBox1: TDBLookupListBox;	public Provious Paulages
Label1: TLabel;	Preview : Boolean;
,	end;
<> procedure	
FormShow(Sender: TObject);	var
> procedure	PickRpt: TPickRpt;
DBLookupListBox1Click(Sender: TObject);	
3 //	implementation
private	
{ Private declarations }	{\$R *.dfm}
public	
{ Public declarations }	procedure
end;	TPickRpt.OKBtnClick(Sender:
	TObject);
var	begin
PickOrderNoDlg:	Preview := False;
TPickOrderNoDlg;	end;

TOrdersByDateReport;
implementation
{\$R *.dfm}
procedure
TOrdersByDateReport.FormCreate(
Sender: TObject);
begin
end;
end.

implementation
uses DataMod;
,
{\$R *.dfm}
(* ******)
procedure
TPickOrderNoDlg.FormShow(Sende
r: TObject);
begin
MastData.Orders.Open;
<u> </u>
end;
nuonaduun
procedure Thick Order No Dig DBL a claum Ligt Ba
TPickOrderNoDlg.DBLookupListBo
x1Click(Sender: TObject);
begin
end;
end.
unit OrderRpt;
interface
uses SysUtils, Windows, Messages.
uses SysUtils, Windows, Messages, Classes, Graphics, Controls,
Classes, Graphics, Controls,
Classes, Graphics, Controls, StdCtrls, ExtCtrls, Forms,
Classes, Graphics, Controls,
Classes, Graphics, Controls, StdCtrls, ExtCtrls, Forms, Quickrpt, QRCtrls;
Classes, Graphics, Controls, StdCtrls, ExtCtrls, Forms, Quickrpt, QRCtrls; type
Classes, Graphics, Controls, StdCtrls, ExtCtrls, Forms, Quickrpt, QRCtrls;  type TOrdersByDateReport =
Classes, Graphics, Controls, StdCtrls, ExtCtrls, Forms, Quickrpt, QRCtrls;  type TOrdersByDateReport = class(TQuickRep)
Classes, Graphics, Controls, StdCtrls, ExtCtrls, Forms, Quickrpt, QRCtrls;  type TOrdersByDateReport = class(TQuickRep) <> procedure
Classes, Graphics, Controls, StdCtrls, ExtCtrls, Forms, Quickrpt, QRCtrls;  type TOrdersByDateReport = class(TQuickRep) <> procedure FormCreate(Sender: TObject);
Classes, Graphics, Controls, StdCtrls, ExtCtrls, Forms, Quickrpt, QRCtrls;  type TOrdersByDateReport = class(TQuickRep) <> procedure
Classes, Graphics, Controls, StdCtrls, ExtCtrls, Forms, Quickrpt, QRCtrls;  type  TOrdersByDateReport = class(TQuickRep) <> procedure FormCreate(Sender: TObject); private
Classes, Graphics, Controls, StdCtrls, ExtCtrls, Forms, Quickrpt, QRCtrls;  type TOrdersByDateReport = class(TQuickRep) <> procedure FormCreate(Sender: TObject);
Classes, Graphics, Controls, StdCtrls, ExtCtrls, Forms, Quickrpt, QRCtrls;  type TOrdersByDateReport = class(TQuickRep) <> procedure FormCreate(Sender: TObject); private  public
Classes, Graphics, Controls, StdCtrls, ExtCtrls, Forms, Quickrpt, QRCtrls;  type  TOrdersByDateReport = class(TQuickRep) <> procedure FormCreate(Sender: TObject); private
Classes, Graphics, Controls, StdCtrls, ExtCtrls, Forms, Quickrpt, QRCtrls;  type TOrdersByDateReport = class(TQuickRep) <> procedure FormCreate(Sender: TObject); private  public
Classes, Graphics, Controls, StdCtrls, ExtCtrls, Forms, Quickrpt, QRCtrls;  type TOrdersByDateReport = class(TQuickRep) <> procedure FormCreate(Sender: TObject); private  public

### <u>AC.&P</u>

BackWall.Color = clWhite Foot.Alignment = taLeftJustify Foot.Font.Charset = DEFAULT_CHARSET Foot.Font.Color = clRed Foot.Font.Height = -12 Foot.Font.Name = 'Arial' Foot.Frame.Color = clScrollBar Foot.Text.Strings = (  'Left drag (Up/Down) to Zoom. Right drag to Scroll. Invert Zoom r' +  'ectangle to reset.') Title.Alignment = taRightJustify Title.Font.Charset = DEFAULT_CHARSET Title.Font.Color = clBlue Title.Font.Height = -15 Title.Font.Name = 'Arial' Title.Font.Style = [fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = (  'This demo shows the Bubble chart Type.'  'Each Point is a Bubble with a custom Radius, Color and Label.') BackColor = clWhite BottomAxis.LabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clTeal BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Color = clGreen	B INVINCIA WALL
Foot.Font.Charset = DEFAULT_CHARSET Foot.Font.Color = clRed Foot.Font.Height = -12 Foot.Font.Name = 'Arial' Foot.Font.Style = [fsItalic] Foot.Frame.Color = clScrollBar Foot.Text.Strings = (  'Left drag (Up/Down) to Zoom. Right drag to Scroll. Invert Zoom r' +  'ectangle to reset.') Title.Alignment = taRightJustify Title.Font.Charset = DEFAULT_CHARSET Title.Font.Color = clBlue Title.Font.Height = -15 Title.Font.Name = 'Arial' Title.Font.Style = [fsItalic] Title.Font.Style = [fsItalic] Title.Text.Strings = (  'This demo shows the Bubble chart Type.'  'Each Point is a Bubble with a custom Radius, Color and Label.') BackColor = clWhite BottomAxis.LabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clTeal BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Height = -	
DEFAULT_CHARSET Foot.Font.Color = clRed Foot.Font.Height = -12 Foot.Font.Name = 'Arial' Foot.Font.Style = [fsItalic] Foot.Frame.Color = clScrollBar Foot.Text.Strings = (  'Left drag (Up/Down) to Zoom. Right drag to Scroll. Invert Zoom r' +  'ectangle to reset.') Title.Alignment = taRightJustify Title.Font.Charset = DEFAULT_CHARSET Title.Font.Height = -15 Title.Font.Name = 'Arial' Title.Font.Style = [fsItalic] Title.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = (  'This demo shows the Bubble chart Type.'  'Each Point is a Bubble with a custom Radius, Color and Label.') BackColor = clWhite BottomAxis.LabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clTeal BottomAxis.LabelsFont.Height = -12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Name = 'Arial'	·
Foot.Font.Color = clRed Foot.Font.Height = -12 Foot.Font.Name = 'Arial' Foot.Font.Style = [fsItalic] Foot.Frame.Color = clScrollBar Foot.Text.Strings = (  'Left drag (Up/Down) to Zoom. Right drag to Scroll. Invert Zoom r' +  'ectangle to reset.') Title.Alignment = taRightJustify Title.Font.Charset = DEFAULT_CHARSET Title.Font.Height = -15 Title.Font.Height = -15 Title.Font.Style = [fsItalic] Title.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = (  'This demo shows the Bubble chart Type.'  'Each Point is a Bubble with a custom Radius, Color and Label.') BackColor = clWhite BottomAxis.LabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clTeal BottomAxis.LabelsFont.Height = -12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Height = -16	
Foot.Font.Name = 'Arial' Foot.Font.Style = [fsItalic] Foot.Frame.Color = clScrollBar Foot.Text.Strings = (  'Left drag (Up/Down) to Zoom. Right drag to Scroll. Invert Zoom r' +  'ectangle to reset.') Title.Alignment = taRightJustify Title.Font.Charset = DEFAULT_CHARSET Title.Font.Height = -15 Title.Font.Name = 'Arial' Title.Font.Style = [fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = (  'This demo shows the Bubble chart Type.'  'Each Point is a Bubble with a custom Radius, Color and Label.') BackColor = clWhite BottomAxis.LabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clTeal BottomAxis.LabelsFont.Height = -12 BottomAxis.LabelsFont.Style = [] BottomAxis.LabelsFont.Color = clTeal BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Style =	
Foot.Font.Name = 'Arial' Foot.Font.Style = [fsItalic] Foot.Frame.Color = clScrollBar Foot.Text.Strings = (  'Left drag (Up/Down) to Zoom. Right drag to Scroll. Invert Zoom r' +  'ectangle to reset.') Title.Alignment = taRightJustify Title.Font.Charset = DEFAULT_CHARSET Title.Font.Height = -15 Title.Font.Height = -15 Title.Font.Style = [fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = (  'This demo shows the Bubble chart Type.'  'Each Point is a Bubble with a custom Radius, Color and Label.') BackColor = clWhite BottomAxis.LabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clTeal BottomAxis.LabelsFont.Height = -12 BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = -16	Foot.Font.Color = clRed
Foot.Font.Style = [fsItalic] Foot.Frame.Color = clScrollBar Foot.Text.Strings = (  'Left drag (Up/Down) to Zoom. Right drag to Scroll. Invert Zoom r' +  'ectangle to reset.') Title.Alignment = taRightJustify Title.Font.Charset = DEFAULT_CHARSET  Title.Font.Color = clBlue Title.Font.Height = -15 Title.Font.Name = 'Arial' Title.Font.Style = [fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = (  'This demo shows the Bubble chart Type.'  'Each Point is a Bubble with a custom Radius, Color and Label.') BackColor = clWhite BottomAxis.LabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clTeal BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Charset = ClGreen BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16	Foot.Font.Height = -12
Foot.Frame.Color = clScrollBar Foot.Text.Strings = (  'Left drag (Up/Down) to Zoom. Right drag to Scroll. Invert Zoom r' +  'ectangle to reset.') Title.Alignment = taRightJustify Title.Font.Charset = DEFAULT_CHARSET Title.Font.Height = -15 Title.Font.Name = 'Arial' Title.Font.Style = [fsItalic] Title.Font.Style = [fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'This demo shows the Bubble chart Type.'  'Each Point is a Bubble with a custom Radius, Color and Label.') BackColor = clWhite BottomAxis.LabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clTeal BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Charset = OEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Name = 'Arial'	Foot.Font.Name = 'Arial'
Foot.Text.Strings = (   'Left drag (Up/Down) to Zoom.   Right drag to Scroll. Invert Zoom r' +     'ectangle to reset.')   Title.Alignment = taRightJustify   Title.Font.Charset =   DEFAULT_CHARSET     Title.Font.Color = clBlue   Title.Font.Height = -15   Title.Font.Name = 'Arial'   Title.Font.Style = [fsItalic]   Title.Frame.Color = clScrollBar   Title.Text.Strings = (	Foot.Font.Style = [fsItalic]
Foot.Text.Strings = (   'Left drag (Up/Down) to Zoom.   Right drag to Scroll. Invert Zoom r' +     'ectangle to reset.')   Title.Alignment = taRightJustify   Title.Font.Charset =   DEFAULT_CHARSET     Title.Font.Color = clBlue   Title.Font.Height = -15   Title.Font.Name = 'Arial'   Title.Font.Style = [fsItalic]   Title.Frame.Color = clScrollBar   Title.Text.Strings = (	Foot.Frame.Color = clScrollBar
'Left drag (Up/Down) to Zoom. Right drag to Scroll. Invert Zoom r' +     'ectangle to reset.') Title.Alignment = taRightJustify Title.Font.Charset = DEFAULT_CHARSET Title.Font.Color = clBlue Title.Font.Height = -15 Title.Font.Name = 'Arial' Title.Font.Style = [fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = (     'This demo shows the Bubble chart Type.'     'Each Point is a Bubble with a custom Radius, Color and Label.') BackColor = clWhite BottomAxis.LabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clTeal BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Name = 'Arial'	
Right drag to Scroll. Invert Zoom r'  +  'ectangle to reset.')  Title.Alignment = taRightJustify  Title.Font.Charset =  DEFAULT_CHARSET  Title.Font.Height = -15  Title.Font.Name = 'Arial'  Title.Font.Style = [fsItalic]  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'This demo shows the Bubble chart Type.'  'Each Point is a Bubble with a custom Radius, Color and Label.')  BackColor = clWhite  BottomAxis.LabelsAngle = 90  BottomAxis.LabelsFont.Charset =  DEFAULT_CHARSET  BottomAxis.LabelsFont.Color = clTeal  BottomAxis.LabelsFont.Height = -  12  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -  16  BottomAxis.Title.Font.Height = -  16  BottomAxis.Title.Font.Name =  'Arial'  BottomAxis.Title.Font.Name =  'Arial'	
Right drag to Scroll. Invert Zoom r'  +  'ectangle to reset.')  Title.Alignment = taRightJustify  Title.Font.Charset =  DEFAULT_CHARSET  Title.Font.Height = -15  Title.Font.Name = 'Arial'  Title.Font.Style = [fsItalic]  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'This demo shows the Bubble chart Type.'  'Each Point is a Bubble with a custom Radius, Color and Label.')  BackColor = clWhite  BottomAxis.LabelsAngle = 90  BottomAxis.LabelsFont.Charset =  DEFAULT_CHARSET  BottomAxis.LabelsFont.Color = clTeal  BottomAxis.LabelsFont.Height = -  12  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -  16  BottomAxis.Title.Font.Height = -  16  BottomAxis.Title.Font.Name =  'Arial'  BottomAxis.Title.Font.Name =  'Arial'	'Left drag (Un/Down) to Zoom.
'ectangle to reset.')  Title.Alignment = taRightJustify  Title.Font.Charset =  DEFAULT_CHARSET  Title.Font.Color = clBlue  Title.Font.Height = -15  Title.Font.Name = 'Arial'  Title.Font.Style = [fsItalic]  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'This demo shows the Bubble chart Type.'  'Each Point is a Bubble with a custom Radius, Color and Label.')  BackColor = clWhite  BottomAxis.LabelsAngle = 90  BottomAxis.LabelsFont.Charset =  DEFAULT_CHARSET  BottomAxis.LabelsFont.Color = clTeal  BottomAxis.LabelsFont.Height = -  12  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -  16  BottomAxis.Title.Font.Height = -  16  BottomAxis.Title.Font.Name =  'Arial'  BottomAxis.Title.Font.Name =  'Arial'	
'ectangle to reset.')  Title.Alignment = taRightJustify  Title.Font.Charset =  DEFAULT_CHARSET  Title.Font.Height = -15  Title.Font.Name = 'Arial'  Title.Font.Style = [fsItalic]  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'This demo shows the Bubble chart Type.'  'Each Point is a Bubble with a custom Radius, Color and Label.')  BackColor = clWhite  BottomAxis.LabelsAngle = 90  BottomAxis.LabelsFont.Charset =  DEFAULT_CHARSET  BottomAxis.LabelsFont.Color = clTeal  BottomAxis.LabelsFont.Height = -12  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Name = 'Arial'	
Title.Alignment = taRightJustify Title.Font.Charset = DEFAULT_CHARSET Title.Font.Color = clBlue Title.Font.Height = -15 Title.Font.Name = 'Arial' Title.Font.Style = [fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'This demo shows the Bubble chart Type.' 'Each Point is a Bubble with a custom Radius, Color and Label.') BackColor = clWhite BottomAxis.LabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clTeal BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Style = [] BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Name = 'Arial'	
Title.Font.Charset = DEFAULT_CHARSET Title.Font.Color = clBlue Title.Font.Height = -15 Title.Font.Name = 'Arial' Title.Font.Style = [fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'This demo shows the Bubble chart Type.' 'Each Point is a Bubble with a custom Radius, Color and Label.') BackColor = clWhite BottomAxis.LabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clTeal BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Charset = OEFAULT_CHARSET BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Charset = OEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Style = BottomAxis.Title.Font.Name = 'Arial'	
Title.Font.Color = clBlue  Title.Font.Height = -15  Title.Font.Name = 'Arial'  Title.Font.Style = [fsItalic]  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'This demo shows the Bubble chart Type.'  'Each Point is a Bubble with a custom Radius, Color and Label.')  BackColor = clWhite  BottomAxis.LabelsAngle = 90  BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET  BottomAxis.LabelsFont.Color = clTeal  BottomAxis.LabelsFont.Height = -12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.Title.Font.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Charset = OEFAULT_CHARSET  BottomAxis.Title.Font.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Charset = OEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style = Interpretation of the properties of the propertie	
Title.Font.Height = -15  Title.Font.Name = 'Arial'  Title.Font.Style = [fsItalic]  Title.Frame.Color = clScrollBar  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'This demo shows the Bubble chart Type.'  'Each Point is a Bubble with a custom Radius, Color and Label.')  BackColor = clWhite  BottomAxis.LabelsAngle = 90  BottomAxis.LabelsFont.Charset =  DEFAULT_CHARSET  BottomAxis.LabelsFont.Color = clTeal  BottomAxis.LabelsFont.Height = - 12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Name = 'Arial'	
Title.Font.Name = 'Arial'  Title.Font.Style = [fsItalic]  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'This demo shows the Bubble chart Type.'  'Each Point is a Bubble with a custom Radius, Color and Label.')  BackColor = clWhite  BottomAxis.LabelsAngle = 90  BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET  BottomAxis.LabelsFont.Color = clTeal  BottomAxis.LabelsFont.Height = -12  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Charset = 16  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Name = 'Arial'	
Title.Font.Name = 'Arial' Title.Font.Style = [fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'This demo shows the Bubble chart Type.' 'Each Point is a Bubble with a custom Radius, Color and Label.') BackColor = clWhite BottomAxis.LabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clTeal BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Charset = 10	
Title.Font.Style = [fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = (     'This demo shows the Bubble chart Type.'     'Each Point is a Bubble with a custom Radius, Color and Label.') BackColor = clWhite BottomAxis.LabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clTeal BottomAxis.LabelsFont.Height = -12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Charset = clGreen BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Style = []	
Title.Frame.Color = clScrollBar Title.Text.Strings = (     'This demo shows the Bubble chart Type.'     'Each Point is a Bubble with a custom Radius, Color and Label.') BackColor = clWhite BottomAxis.LabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clTeal BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Style =	
Title.Text.Strings = (     'This demo shows the Bubble chart Type.'      'Each Point is a Bubble with a custom Radius, Color and Label.')  BackColor = clWhite  BottomAxis.LabelsAngle = 90  BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET  BottomAxis.LabelsFont.Color = clTeal  BottomAxis.LabelsFont.Height = -12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.Title.Font.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style = []	
'This demo shows the Bubble chart Type.'  'Each Point is a Bubble with a custom Radius, Color and Label.')  BackColor = clWhite  BottomAxis.LabelsAngle = 90  BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET  BottomAxis.LabelsFont.Color = clTeal  BottomAxis.LabelsFont.Height = -12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.Title.Font.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style = []	
'Each Point is a Bubble with a custom Radius, Color and Label.')  BackColor = clWhite  BottomAxis.LabelsAngle = 90  BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET  BottomAxis.LabelsFont.Color = clTeal  BottomAxis.LabelsFont.Height = - 12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style =	<u> </u>
'Each Point is a Bubble with a custom Radius, Color and Label.')  BackColor = clWhite  BottomAxis.LabelsAngle = 90  BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET  BottomAxis.LabelsFont.Height = - 12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style =	'This demo shows the Bubble
custom Radius, Color and Label.')  BackColor = clWhite  BottomAxis.LabelsAngle = 90  BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET  BottomAxis.LabelsFont.Color = clTeal  BottomAxis.LabelsFont.Height = -12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.Title.Font.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style =	chart Type.'
BackColor = clWhite BottomAxis.LabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clTeal BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Style =	'Each Point is a Bubble with a
BottomAxis.LabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clTeal BottomAxis.LabelsFont.Height = -12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Style =	custom Radius, Color and Label.')
BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET  BottomAxis.LabelsFont.Color = clTeal  BottomAxis.LabelsFont.Height = -12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style =	BackColor = clWhite
BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET  BottomAxis.LabelsFont.Color = clTeal  BottomAxis.LabelsFont.Height = -12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style =	BottomAxis.LabelsAngle = 90
DEFAULT_CHARSET  BottomAxis.LabelsFont.Color = clTeal  BottomAxis.LabelsFont.Height = -12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style =	
BottomAxis.LabelsFont.Color = clTeal  BottomAxis.LabelsFont.Height = -12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style =	
BottomAxis.LabelsFont.Height = - 12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style =	
BottomAxis.LabelsFont.Height = - 12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style =	
BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style =	
BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style =	
'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Color =  clGreen  BottomAxis.Title.Font.Height = -  16  BottomAxis.Title.Font.Name =  'Arial'  BottomAxis.Title.Font.Style =	
BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style =	
BottomAxis.Title.Font.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style =	
DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style =	
BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style =	
ClGreen  BottomAxis.Title.Font.Height = -  16  BottomAxis.Title.Font.Name =  'Arial'  BottomAxis.Title.Font.Style =	<del>-</del>
BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style =	
BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Style =	
BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Style =	
'Arial' BottomAxis.Title.Font.Style =	
BottomAxis.Title.Font.Style =	
· ·	
[fsItalic]	
	[fsItalic]

Objects643	
الهلف لبرناهج	
(TeeChart)	
إعداد	
علاء الدين اللباد	
واتف ۷۳۷۱	

برنامج يتعلق بالرسوم البيانية والإحصائية يحتوي العديد من الميزات الرائعة في برنامج دلفي ٧ وهو موجود كمثال باسم ضمن البرنامج نفسه ويمكن الاستفادة من جميع الميزات الموجودة فيه قام بتحليل وشرح البرنامج علاء الدين اللباد ١٩٤١٥٧٥٣١، الملف النصي للبرنامج ويحتوي على ١٤٣٠ عنصرا الملف النصي للبرنامج ويحتوي على ٥٤٤٥ كافورا ويحتوي على ٥٤٤٠ كافورا وجميعها معلمة بالسهم وجميعها معلمة بالسهم

# <u>AC.&P</u>

Pointer.HorizSize = 57
Pointer.InflateMargins = False
Pointer.Style = psCircle
Pointer.VertSize = 57
Pointer.Visible = True
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.0000000000000000000
YValues.Order = loNone
RadiusValues.DateTime = False
RadiusValues.Name = 'Radius'
RadiusValues.Multiplier =
1.00000000000000000000
RadiusValues.Order = loNone
end
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 153
Height = 405
Align = alLeft
TabOrder = 1
<> object Label1: TLabel
Left = 6
Top = 37
Width = 72
Height = 17
HelpType = htKeyword
Alignment = taRightJustify
AutoSize = False
Caption = 'Bubble &Style:'
end
<> object ZoomInButton:
=
TSpeedButton Left = 40
Top = 328
Width = 25
Height = 25
HelpType = htKeyword
Glyph.Data = {
NumGlyphs = 2

LeftAxis.LabelsFont.Charset =
DEFAULT_CHARSET
LeftAxis.LabelsFont.Color =
clWhite
LeftAxis.LabelsFont.Height = -13
LeftAxis.LabelsFont.Name =
'Arial'
LeftAxis.LabelsFont.Style = []
LeftAxis.MinorTickLength = 4 LeftAxis.MinorTicks.Color =
4227327
LeftAxis.TickLength = 8
LeftAxis.Title.Angle = 0
LeftAxis.Title.Caption = 'Stock
Price'
LeftAxis.Title.Font.Charset =
DEFAULT CHARSET
LeftAxis.Title.Font.Color =
clNavy
LeftAxis.Title.Font.Height = -15
LeftAxis.Title.Font.Name = 'Arial'
LeftAxis.Title.Font.Style =
[fsBold]
Legend.ColorWidth = 32
Legend.Visible = False
RightAxis.Title.Angle = 180
TopAxis.Title.Caption = 'Date'
View3D = False
Align = alClient
BevelWidth = 0
TabOrder = 0
<> object BubbleSeries1:
TBubbleSeries
HorizAxis = aTopAxis
Marks.ArrowLength = 0 Marks.Clip = True
Marks.Cnp - True  Marks.Font.Charset =
DEFAULT CHARSET
Marks.Font.Color = clWhite
Marks.Font.Height = -16
Marks.Font.Name = 'Arial'
Marks.Font.Style = [fsItalic]
Marks.Frame.Color = 8454143
Marks.Frame.Visible = False
Marks.Transparent = True
Marks. Visible = True
SeriesColor = clRed
OnGetPointerStyle =
BubbleSeries1GetPointerStyle
·

Style = csDropDownList
ItemHeight = 13
TabOrder = 2
OnChange = ComboBox1Change
Items.Strings = (
'Rectangle'
'Circle'
'Triangle'
'DownTriangle'
'Cross'
'DiagCross'
'Star'
'Diamond')
end
<> object BitBtn3: TBitBtn
Left = 28
Top = 364
Width = 89
Height = 33
TabOrder = 3
Kind = bkClose
end
<> object CheckBox3:
TCheckBox
Left = 16
Top = 88
Width = 97
Height = 17
Caption = '&Random'
TabOrder = 4
OnClick = CheckBox3Click
end
<> object Memo1: TMemo
Left = 16
Top = 136
Width = 121
Height = 185
Lines.Strings = (
'Each "Bubble" is '
'represented with a '
'center position and '
'Radius.'
'Each point can have a '
'different style using the '
'OnGetPointerStyle '
'event.'
"
'This example also '

OnClick = ZoomInButtonClick
end
<> object ZoomOutButton:
TSpeedButton
Left = 72
Top = 328
Width = 25
Height = 25
HelpType = htKeyword
Glyph.Data = {
NumGlyphs = 2
V 1
OnClick = ZoomOutButtonClick
end
<> object CheckBox1:
TCheckBox
Left = 17
Top = 8
Width = 128
Height = 17
Caption = '&Animate !!!'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -19
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 0
OnClick = CheckBox1Click
end
<> object CheckBox2:
TCheckBox
Left = 16
Top = 104
Width = 93
Height = 17
Caption = 'View & Marks'
Checked = True
State = cbChecked
TabOrder = 1
OnClick = CheckBox2Click
end
<> object ComboBox1:
TComboBox
Left = 14
Top = 53
Width = 131
Height = 21

Foot.Font.Style = [fsItalic] Foot.Frame.Color = clScrollBar Foot.Text.Strings = (  ''  '') Foot.Visible = False LeftWall.Color = 16777088 LeftWall.Size = 5 MarginBottom = 7 MarginLeft = 5 Title.Font.Charset = DEFAULT_CHARSET Title.Font.Color = clGreen Title.Font.Height = -16 Title.Font.Style = [fsBold, fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = (  'Line Series') BottomAxis.Axis.Width = 1 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Style = [fsItalic] Chart3DPercent = 30 LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Color = clPurple	Food Fond Names — ! A right
Foot.Frame.Color = clScrollBar Foot.Text.Strings = (  ''  '') Foot.Visible = False LeftWall.Color = 16777088 LeftWall.Size = 5 MarginBottom = 7 MarginLeft = 5 Title.Font.Charset = DEFAULT_CHARSET Title.Font.Height = -16 Title.Font.Name = 'Arial' Title.Font.Style = [fsBold, fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = (  'Line Series') BottomAxis.Axis.Width = 1 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Style = [] BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Style = [fsItalic] Chart3DPercent = 30 LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Color = clPurple	Foot.Font.Name = 'Arial'
Foot.Text.Strings = (  ''  '')  Foot.Visible = False  LeftWall.Color = 16777088  LeftWall.Size = 5  MarginBottom = 7  MarginLeft = 5  Title.Font.Charset =  DEFAULT_CHARSET  Title.Font.Height = -16  Title.Font.Name = 'Arial'  Title.Font.Style = [fsBold, fsItalic]  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'Line Series')  BottomAxis.Axis.Width = 1  BottomAxis.LabelsFont.Charset =  DEFAULT_CHARSET  BottomAxis.LabelsFont.Height = -  12  BottomAxis.LabelsFont.Height = -  12  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Color = clGretAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET  LeftAxis.LabelsFont.Color = clPurple	
''  '')  Foot.Visible = False  LeftWall.Color = 16777088  LeftWall.Size = 5  MarginBottom = 7  MarginLeft = 5  Title.Font.Charset =  DEFAULT CHARSET  Title.Font.Height = -16  Title.Font.Style = [fsBold, fsItalic]  Title.Font.Style = [fsBold, fsItalic]  Title.Fame.Color = clScrollBar  Title.Text.Strings = (  'Line Series')  BottomAxis.Axis.Width = 1  BottomAxis.LabelsFont.Charset =  DEFAULT_CHARSET  BottomAxis.LabelsFont.Height = -  12  BottomAxis.LabelsFont.Height = -  12  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Style = []  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Color = clGretAult_Charset = DEFAULT_Charset = DEFAUL	
'') Foot.Visible = False LeftWall.Color = 16777088 LeftWall.Size = 5 MarginBottom = 7 MarginLeft = 5 Title.Font.Charset = DEFAULT_CHARSET Title.Font.Height = -16 Title.Font.Name = 'Arial' Title.Font.Style = [fsBold, fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'Line Series') BottomAxis.Axis.Width = 1 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Height = -12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Name = 'Arial'	
Foot.Visible = False LeftWall.Color = 16777088 LeftWall.Size = 5 MarginBottom = 7 MarginLeft = 5 Title.Font.Charset = DEFAULT_CHARSET Title.Font.Height = -16 Title.Font.Name = 'Arial' Title.Font.Style = [fsBold, fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'Line Series') BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clBlue BottomAxis.LabelsFont.Height = -12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Color = clGretAxis.Axis.Width = 1 LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET LeftAxis.LabelsFont.Color = clPurple	
LeftWall.Color = 16777088 LeftWall.Size = 5 MarginBottom = 7 MarginLeft = 5 Title.Font.Charset = DEFAULT_CHARSET Title.Font.Height = -16 Title.Font.Name = 'Arial' Title.Font.Style = [fsBold, fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'Line Series') BottomAxis.Axis.Width = 1 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Height = -12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Charset = [] BottomAxis.Title.Font.Charset = [] BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Style = [] BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Color = clGreat3DPercent = 30 LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET LeftAxis.LabelsFont.Color = clPurple	/
LeftWall.Size = 5 MarginBottom = 7 MarginLeft = 5 Title.Font.Charset = DEFAULT_CHARSET Title.Font.Height = -16 Title.Font.Name = 'Arial' Title.Font.Style = [fsBold, fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'Line Series') BottomAxis.Axis.Width = 1 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Charset = 'Arial' BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Style = [] Chart3DPercent = 30 LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET LeftAxis.LabelsFont.Color = clPurple	
MarginBottom = 7 MarginLeft = 5 Title.Font.Charset = DEFAULT_CHARSET Title.Font.Height = -16 Title.Font.Name = 'Arial' Title.Font.Style = [fsBold, fsItalic] Title.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = (     'Line Series') BottomAxis.Axis.Width = 1 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Style = [fsItalic] Chart3DPercent = 30 LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET LeftAxis.LabelsFont.Color = clPurple	LeftWall.Color = 16777088
MarginLeft = 5 Title.Font.Charset = DEFAULT_CHARSET Title.Font.Color = clGreen Title.Font.Height = -16 Title.Font.Name = 'Arial' Title.Font.Style = [fsBold, fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = (     'Line Series') BottomAxis.Axis.Width = 1 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Style = [ glGreen BottomAxis.Title.Font.Height = - 16 LeftAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Style = [ fsItalic] Chart3DPercent = 30 LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET LeftAxis.LabelsFont.Color = clPurple	LeftWall.Size = 5
MarginLeft = 5 Title.Font.Charset = DEFAULT_CHARSET Title.Font.Color = clGreen Title.Font.Height = -16 Title.Font.Name = 'Arial' Title.Font.Style = [fsBold, fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = (     'Line Series') BottomAxis.Axis.Width = 1 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Style = [ glGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Color = clGreen LeftAxis.Title.Font.Style = [ fsItalic] Chart3DPercent = 30 LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET LeftAxis.LabelsFont.Color = clPurple	MarginBottom = 7
Title.Font.Charset = DEFAULT CHARSET Title.Font.Color = clGreen Title.Font.Height = -16 Title.Font.Name = 'Arial' Title.Font.Style = [fsBold, fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = (     'Line Series') BottomAxis.Axis.Width = 1 BottomAxis.LabelsFont.Charset = DEFAULT CHARSET BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Charset = DEFAULT CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Style = [ Fistalic] Chart3DPercent = 30 LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Color = clPurple	
DEFAULT CHARSET  Title.Font.Color = clGreen  Title.Font.Height = -16  Title.Font.Name = 'Arial'  Title.Font.Style = [fsBold, fsItalic]  Title.Frame.Color = clScrollBar  Title.Text.Strings = (     'Line Series')  BottomAxis.Axis.Width = 1  BottomAxis.LabelsFont.Charset = DEFAULT CHARSET  BottomAxis.LabelsFont.Height = -12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset = DEFAULT CHARSET  BottomAxis.Title.Font.Charset = DEFAULT CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style = [fsItalic]  Chart3DPercent = 30  LeftAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset = DEFAULT CHARSET  LeftAxis.LabelsFont.Color = clPurple	
Title.Font.Color = clGreen Title.Font.Height = -16 Title.Font.Name = 'Arial' Title.Font.Style = [fsBold, fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = (     'Line Series') BottomAxis.Axis.Width = 1 BottomAxis.LabelsFont.Charset = DEFAULT CHARSET BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Charset = DEFAULT CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Style = [fsItalic] Chart3DPercent = 30 LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET LeftAxis.LabelsFont.Color = clPurple	
Title.Font.Name = 'Arial' Title.Font.Style = [fsBold, fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = (     'Line Series') BottomAxis.Axis.Width = 1 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clBlue BottomAxis.LabelsFont.Height = -12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Style = [fsItalic] Chart3DPercent = 30 LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Color = clPurple	_
Title.Font.Name = 'Arial' Title.Font.Style = [fsBold, fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = (     'Line Series') BottomAxis.Axis.Width = 1 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Style = [fsItalic] Chart3DPercent = 30 LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Color = clPurple	
Title.Font.Style = [fsBold, fsItalic] Title.Frame.Color = clScrollBar Title.Text.Strings = (     'Line Series') BottomAxis.Axis.Width = 1 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clBlue BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Style = [fsItalic] Chart3DPercent = 30 LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET LeftAxis.LabelsFont.Color = clPurple	C
Title.Frame.Color = clScrollBar Title.Text.Strings = (     'Line Series') BottomAxis.Axis.Width = 1 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clBlue BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Style = [fsItalic] Chart3DPercent = 30 LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET LeftAxis.LabelsFont.Color = clPurple	
Title.Text.Strings = (     'Line Series')  BottomAxis.Axis.Width = 1  BottomAxis.LabelsFont.Charset = DEFAULT CHARSET  BottomAxis.LabelsFont.Color = clBlue  BottomAxis.LabelsFont.Height = - 12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset = DEFAULT CHARSET  BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Style = [ fsltalic]  Chart3DPercent = 30  LeftAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset = DEFAULT CHARSET  LeftAxis.LabelsFont.Color = clPurple	
'Line Series')  BottomAxis.Axis.Width = 1  BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET  BottomAxis.LabelsFont.Color = clBlue  BottomAxis.LabelsFont.Height = - 12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style = [fsItalic]  Chart3DPercent = 30  LeftAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET  LeftAxis.LabelsFont.Color = clPurple	
BottomAxis.Axis.Width = 1 BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET BottomAxis.LabelsFont.Color = clBlue BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Style = [fsItalic] Chart3DPercent = 30 LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET LeftAxis.LabelsFont.Color = clPurple	9 \
BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET  BottomAxis.LabelsFont.Color = clBlue BottomAxis.LabelsFont.Height = - 12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style = [fsItalic] Chart3DPercent = 30 LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET  LeftAxis.LabelsFont.Color = clPurple	
DEFAULT CHARSET  BottomAxis.LabelsFont.Color = clBlue  BottomAxis.LabelsFont.Height = -12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Charset = DEFAULT CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style = [fsItalic]  Chart3DPercent = 30  LeftAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset = DEFAULT CHARSET  LeftAxis.LabelsFont.Color = clPurple	
BottomAxis.LabelsFont.Color = clBlue  BottomAxis.LabelsFont.Height = -12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style = [fsItalic]  Chart3DPercent = 30  LeftAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET  LeftAxis.LabelsFont.Color = clPurple	
BottomAxis.LabelsFont.Height = - 12  BottomAxis.LabelsFont.Name =  'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Color =  clGreen  BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Name =  'Arial'  BottomAxis.Title.Font.Style =  [fsItalic]  Chart3DPercent = 30  LeftAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset =  DEFAULT_CHARSET  LeftAxis.LabelsFont.Color =  clPurple	_
BottomAxis.LabelsFont.Height = - 12  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset = DEFAULT CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style = [fsItalic]  Chart3DPercent = 30  LeftAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset = DEFAULT CHARSET  LeftAxis.LabelsFont.Color = clPurple	
BottomAxis.LabelsFont.Name =  'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Color =  clGreen  BottomAxis.Title.Font.Height = -  16  BottomAxis.Title.Font.Name =  'Arial'  BottomAxis.Title.Font.Style =  [fsItalic]  Chart3DPercent = 30  LeftAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset =  DEFAULT_CHARSET  LeftAxis.LabelsFont.Color =  clPurple	
BottomAxis.LabelsFont.Name =  'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Color =  clGreen  BottomAxis.Title.Font.Height = -  16  BottomAxis.Title.Font.Name =  'Arial'  BottomAxis.Title.Font.Style =  [fsItalic]  Chart3DPercent = 30  LeftAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset =  DEFAULT_CHARSET  LeftAxis.LabelsFont.Color =  clPurple	
'Arial' BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Name = 'Arial' BottomAxis.Title.Font.Style = [fsItalic] Chart3DPercent = 30 LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET LeftAxis.LabelsFont.Color = clPurple	
BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Font.Charset =  DEFAULT_CHARSET  BottomAxis.Title.Font.Color =  clGreen  BottomAxis.Title.Font.Height = -  16  BottomAxis.Title.Font.Name =  'Arial'  BottomAxis.Title.Font.Style =  [fsItalic]  Chart3DPercent = 30  LeftAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset =  DEFAULT_CHARSET  LeftAxis.LabelsFont.Color =  clPurple	BottomAxis.LabelsFont.Name =
BottomAxis.Title.Font.Charset = DEFAULT CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style = [fsItalic]  Chart3DPercent = 30  LeftAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset = DEFAULT CHARSET  LeftAxis.LabelsFont.Color = clPurple	'Arial'
DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style = [fsItalic]  Chart3DPercent = 30  LeftAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET  LeftAxis.LabelsFont.Color = clPurple	BottomAxis.LabelsFont.Style = []
BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style = [fsItalic]  Chart3DPercent = 30  LeftAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET  LeftAxis.LabelsFont.Color = clPurple	BottomAxis.Title.Font.Charset =
ClGreen  BottomAxis.Title.Font.Height = -  16  BottomAxis.Title.Font.Name =  'Arial'  BottomAxis.Title.Font.Style =  [fsItalic]  Chart3DPercent = 30  LeftAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset =  DEFAULT_CHARSET  LeftAxis.LabelsFont.Color =  clPurple	DEFAULT CHARSET
BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style = [fsItalic]  Chart3DPercent = 30  LeftAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET  LeftAxis.LabelsFont.Color = clPurple	BottomAxis.Title.Font.Color =
BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style = [fsItalic]  Chart3DPercent = 30  LeftAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset =  DEFAULT_CHARSET  LeftAxis.LabelsFont.Color = clPurple	clGreen
BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style = [fsItalic]  Chart3DPercent = 30  LeftAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset =  DEFAULT_CHARSET  LeftAxis.LabelsFont.Color = clPurple	BottomAxis.Title.Font.Height = -
BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Style = [fsItalic]  Chart3DPercent = 30  LeftAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset =  DEFAULT_CHARSET  LeftAxis.LabelsFont.Color = clPurple	
'Arial' BottomAxis.Title.Font.Style = [fsItalic] Chart3DPercent = 30 LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET LeftAxis.LabelsFont.Color = clPurple	
BottomAxis.Title.Font.Style = [fsItalic] Chart3DPercent = 30 LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET LeftAxis.LabelsFont.Color = clPurple	
[fsItalic] Chart3DPercent = 30 LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET LeftAxis.LabelsFont.Color = clPurple	
Chart3DPercent = 30  LeftAxis.Axis.Width = 1  LeftAxis.LabelsFont.Charset =  DEFAULT_CHARSET  LeftAxis.LabelsFont.Color =  clPurple	
LeftAxis.Axis.Width = 1 LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET LeftAxis.LabelsFont.Color = clPurple	
LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET LeftAxis.LabelsFont.Color = clPurple	
DEFAULT_CHARSET  LeftAxis.LabelsFont.Color = clPurple	
LeftAxis.LabelsFont.Color = clPurple	
clPurple	_
*	
LeftAxis.LabelsFont.Height = $-13$	*
	LeftAxis.LabelsFont.Height = -13
LeftAxis.LabelsFont.Name =	
'Arial'	'Arial'

'shows '
'Axis labels and title '
'rotation.')
TabOrder = 5
end
end
<> object Timer1: TTimer
Enabled = False
Interval = 1
OnTimer = Timer1Timer
Left = 170
Top = 86
end
End
\\\\\\
<> object BasicForm:
TBasicForm
Left = 158
Top = 114
Width = 746
Height = 487
Caption = 'TeeChart Basic Series
Demo'
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
WindowState = wsMaximized
OnCreate = FormCreate
OnResize = FormResize
OnShow = FormShow PixelsPerInch = 96
TextHeight = 13
<> object Chart1: TChart
Left = 128
Top = 49
Width = 329
Height = 200
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BottomWall.Color = 16777088
BottomWall.Size = 6
Foot.Alignment = taLeftJustify
Foot.Font.Charset =
DEFAULT_CHARSET
Foot.Font.Color = clRed
Foot.Font.Height = -12

Marks.Visible = False
SeriesColor = clBlue
LinePen.Color = clBlue
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier = 1.00000000000000000000000000000000000
YValues.Order = loNone
end
end
<> object Chart2: TChart
Left = 128
Top = 256
Width = 329
Height = 173
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
Foot.Alignment = taLeftJustify
Foot.Font.Charset =
DEFAULT_CHARSET
Foot.Font.Color = clBlack
Foot.Font.Height = -12
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsItalic]
Foot.Frame.Color = clScrollBar
Foot.Text.Strings = (
")
Foot.Visible = False
Gradient.Direction = gdRightLeft
Gradient.EndColor = clWhite
Gradient.StartColor = clYellow
LeftWall.Color = clWhite
MarginLeft = 7
MarginRight = 7
MarginTop = 5
Title.Font.Charset =
DEFAULT_CHARSET
Title.Font.Color = clBlue
Title.Font.Height = -16
Title.Font.Name = 'Arial'

<del></del>
LeftAxis.LabelsFont.Style = []
LeftAxis.Title.Font.Charset =
DEFAULT_CHARSET
LeftAxis.Title.Font.Color =
clNavy
LeftAxis.Title.Font.Height = -15
LeftAxis.Title.Font.Name = 'Arial'
LeftAxis.Title.Font.Style =
[fsBold]
Legend.Alignment = laBottom
Legend.Color = clBlack
Legend.ColorWidth = 16
Legend.Font.Charset =
DEFAULT CHARSET
Legend.Font.Color = clWhite
Legend.Font.Height = -12
Legend.Font.Name = 'Arial'
Legend.Font.Style = [fsItalic]
Legend.ShadowColor = clGray
RightAxis.Visible = False
TopAxis.Visible = False
BorderWidth = 1
Color = 8454016
TabOrder = 0
<> object Winter:
TLineSeries
Marks.ArrowLength = 8
Marks.Frame.Color = 8454143
Marks.Visible = False
SeriesColor = clRed
LinePen.Color = clRed
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.00000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Nultiplier =
1.000000000000000000000000000000000000
YValues.Order = loNone
end
<> object Summer:
TLineSeries
Marks.ArrowLength = 8
Marks.Frame.Color = 8454143

TopAxis.LabelsFont.Charset =	T	TOTAL TO A COLL TO DELLOTATE IN I
Title.Text.Strings = (   TopAxis.LabelsFont.Color =	TopAxis.LabelsAngle = 90 TopAxis.LabelsEagle Charget =	Title Frame Color = also rollPar
TopAxis.LabelsFont.Color = clRed		
elRed TopAxis.LabelsFont.Height = -11 TopAxis.LabelsFont.Name = 'Arial' TopAxis.LabelsFont.Style =    TopAxis.LabelsFont.Style =    TopAxis.LabelsFont.Style =    TopAxis.LabelsSize = 30 TabOrder = 3  <> object BarSeries1: TBarSeries Marks.BackColor = 16777088 Marks.Font.Charset = DEFAULT CHARSET BarBult CHARSET Marks.Font.Charset = DEFAULT CHARSET Marks.Font.Charset = DEFAULT CHARSET Marks.Font.Name = 'MS Serif' Marks.Font.Name = 'MS Serif' Marks.Font.Style =    Marks.Style = smsPercent Marks.Visible = True SeriesColor = clRed BarPen.Visible = Tale MultiBar = mbNone XValues.DateTime = True XValues.Multiplier = Lo00000000000000000000000000000000000	_	
TopAxis.LabelsFont.Name = 'Arial' TopAxis.LabelsFont.Style = [] TopAxis.LabelsSize = 30 TabOrder = 3 <> object BarSeries1: TBarSeries Marks.ArrowLength = 20 Marks.BackColor = 16777088 Marks.Font.Charset = DEFAULT CHARSET Marks.Font.Color = clBlack Marks.Font.Color = clBlack Marks.Font.Name = 'MS Scrif' Marks.Font.Style = [] Marks.Font.Style = [] Marks.Style = smsPercent Marks.Style = smsPercent Marks.Visible = True ScriesColor = clRed BarPen.Visible = False MultiBar = mbNone XValues.DateTime = True XValues.Multiplier = 1.0000000000000000000 YValues.Order = loAscending YValues.Multiplier = 1.0000000000000000000 YValues.Order = loAscending YValues.DateTime = False YValues.Multiplier = 1.000000000000000000000000000000000000	<u>-</u>	,
TopAxis.LabelsFont.Name =   'Arial'     TopAxis.LabelsSize = 30     TabOrder = 3     <		
'Arial'   TopAxis.LabelsFont.Style =    TopAxis.LabelsFint.Style =    TopAxis.LabelsSize = 30		
TopAxis.LabelsFont.Style =    TopAxis.LabelsSize = 30 TabOrder = 3 <> object BarSeries1: TBarSeries  Marks.Font.Charset =   DEFAULT CHARSET   BottomAxis.LabelsFont.Height = -11  BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Style =    BottomAxis.LabelsFont.Style =    BottomAxis.LabelsFont.Style =    BottomAxis.LabelsFont.Charset =   DEFAULT CHARSET   BottomAxis.LabelsFont.Charset =   BottomAxis.Title.Font.Charset =   DEFAULT CHARSET   BottomAxis.LabelsFont.Style =    BottomAxis.Title.Font.Charset =   DEFAULT CHARSET   BottomAxis.LabelsFont.Style =    BottomAxis.Title.Font.Charset =   DEFAULT CHARSET   BottomAxis.LabelsFont.Charset =   DEFAULT CHARSET   BottomAxis.LabelsFont.Charset =   DEFAULT CHARSET   BottomAxis.LabelsFont.Charset =   DEFAULT CHARSET   BottomAxis.LabelsFont.Charset =   DEFAULT CHARSET   BottomAxis.Title.Font.Charset =   DEFAULT CHARSET   BottomAxis.LabelsFont.Style =   IbertomAxis.LabelsFont.Style =   BottomAxis.LabelsFont.Charset =   DEFAULT CHARSET   BottomAxis.LabelsFont.Name = 'Arial'   BottomAxis.LabelsFont.Charset =   DEFAULT CHARSET   BottomAxis.LabelsFont.Charset =   DEFAULT CHARSET   BottomAxis.LabelsFont.Charset =   DEFAULT CHARSET   BottomAxis.LabelsFont.Charset =   DEFAULT CHARSET   BottomAxis.LabelsFont.Charset =   ItefAxis.LabelsFont.Charset =   ItefAxis.LabelsFont.Charset =   ItefAxis.LabelsFont.Charset =   ItefAxis.LabelsFont.Charset =   ItefAxis.LabelsFont.Charset =   ItefAxis.LabelsFont.C	-	
TopAxis.LabelsSize = 30 TabOrder = 3  <> object BarScries1: TBarSeries  Marks.BackColor = 16777088  Marks.Font.Charset = DEFAULT CHARSET  Marks.Font.Name = 'MS Scrif' Marks.Font.Style = [] Marks.Style = smsPercent Marks.Visible = True ScriesColor = clRed BarPen.Visible = False MultiBar = mbNone XValues.DateTime = True XValues.Multiplier = 1.000000000000000000000000000000000000		
ClBlue   BottomAxis.LabelsFont.Height = -11	T V	_
BottomAxis.LabelsFont.Height = -11	-	
TBarSeries		
Marks.ArrowLength = 20	<u> </u>	_
Marks.BackColor = 16777088     Marks.Font.Charset =     DEFAULT CHARSET     Marks.Font.Name = 'Image     Marks.Font.Name = 'MS Serif'     Marks.Font.Style =        Marks.Font.Name = 'MS Serif'     Marks.Font.Style =        Marks.Font.Name = 'MS Serif'     Marks.Font.Name = 'MS Serif'     Marks.Font.Style =        BottomAxis.Title.Font.Charset =     DEFAULT CHARSET     BottomAxis.Title.Font.Height = -16     BottomAxis.Title.Font.Color = clBlue     Chart3DPercent = 80     LeftAxis.LabelsFont.Color = clBlue     LeftAxis.Title.Font.Color = clGray		
Marks.Font.Charset = DEFAULT_CHARSET  Marks.Font.Color = clBlack  Marks.Font.Name = 'MS Serif' Marks.Font.Style = []  Marks.Font.Color = clBlack  Marks.Font.Style = []  Marks.Font.Color = clBlack  Marks.Font.Style = []  Marks.Font.Color = clBlack  Marks.Font.Color = clBlue  BottomAxis.Title.Font.Color = clBlue  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Color = clBlue  LeftAxis.LabelsFont.Color = clBlue  LeftAxis		
DEFAULT CHARSET  Marks.Font.Color = clBlack  Marks.Font.Height = -8  Marks.Font.Name = 'MS Serif'  Marks.Font.Style = []  Marks.Style = smsPercent  Marks.Visible = True  SeriesColor = clRed  BarPen.Visible = False  MultiBar = mbNone  XValues.DateTime = True  XValues.Name = 'X'  XValues.Multiplier =  1.00000000000000000000000000000000000		
Marks.Font.Color = clBlack   Marks.Font.Height = -8   Marks.Font.Name = 'MS Serif'   Marks.Font.Style = []   Marks.Style = smsPercent   Marks.Visible = True   SeriesColor = clRed   BarPen.Visible = False   MultiBar = mbNone   XValues.DateTime = True   XValues.Multiplier =   1.00000000000000000000000000000000000		
Marks.Font.Height = -8  Marks.Font.Name = 'MS Serif' Marks.Font.Style = [] Marks.Style = smsPercent Marks.Visible = True SeriesColor = clRed BarPen.Visible = False MultiBar = mbNone XValues.DateTime = True XValues.Multiplier = 1.000000000000000000000000000000000000		-
Marks.Font.Name = 'MS Serif'  Marks.Font.Style = []  Marks.Style = smsPercent  Marks.Visible = True  SeriesColor = clRed  BarPen.Visible = False  MultiBar = mbNone  XValues.DateTime = True  XValues.Multiplier =  1.00000000000000000000000000000000000		
Marks.Font.Style =     Marks.Style = smsPercent  Marks.Visible = True  SeriesColor = clRed  BarPen.Visible = False  MultiBar = mbNone  XValues.DateTime = True  XValues.Name = 'X'  XValues.Multiplier =  1.0000000000000000000  XValues.DateTime = False  YValues.Order = loAscending  YValues.Name = 'Y'  YValues.Multiplier =  1.00000000000000000000000000000000000		
Marks.Style = smsPercent  Marks.Visible = True  SeriesColor = clRed  BarPen.Visible = False  MultiBar = mbNone  XValues.DateTime = True  XValues.Multiplier =  1.00000000000000000  XValues.Order = loAscending  YValues.Multiplier =  1.00000000000000000000000000000000000		_
Marks.Visible = True  SeriesColor = clRed  BarPen.Visible = False  MultiBar = mbNone  XValues.DateTime = True  XValues.Name = 'X'  XValues.Multiplier =  1.00000000000000000000000000000000000		
SeriesColor = clRed   BarPen.Visible = False   MultiBar = mbNone   XValues.DateTime = True   XValues.Name = 'X'   BottomAxis.Title.Font.Style =   [fsItalic]   Chart3DPercent = 80   LeftAxis.Axis.Width = 1   LeftAxis.LabelsFont.Charset =   DEFAULT_CHARSET   LeftAxis.LabelsFont.Color = clBlue   LeftAxis.LabelsFont.Name = 'Arial'   LeftAxis.LabelsFont.Name = 'Arial'   LeftAxis.LabelsFont.Style =   [fsBold]   LeftAxis.Title.Font.Color = clGray   LeftAxis.Title.Font.Height = -15   LeftAxis.Title.Font.Height = -15   LeftAxis.Title.Font.Name = 'Arial'   LeftAxis.Title.Font.Height = -15   LeftAxis.Title.Font.Height = -15   LeftAxis.Title.Font.Style =   [fsBold]   LeftAxis.Title.Font.Height = -15   LeftAxis.Title.Font.Style =   [fsBold]   LeftAxis.Title.Font.Style =   [fsBold]   LeftAxis.Title.Font.Height = -15   LeftAxis.Title.Font.Style =   [fsBold]   LeftAxis.Title.Font.Style =   [fsBold]   LeftAxis.Title.Font.Height = -15   LeftAxis.Title.Font.Style =   [fsBold]   Left		
BarPen.Visible = False  MultiBar = mbNone  XValues.DateTime = True  XValues.Multiplier =  1.00000000000000000000000000000000000		
MultiBar = mbNone    XValues.DateTime = True    XValues.Name = 'X'    XValues.Multiplier =    1.000000000000000000    XValues.Order = loAscending    YValues.DateTime = False    YValues.Name = 'Y'    YValues.Multiplier =    1.0000000000000000000000000000000000		
XValues.DateTime = True  XValues.Name = 'X'  XValues.Multiplier =  1.00000000000000000000000000000000000		
XValues.Name = 'X'  XValues.Multiplier =  1.00000000000000000000000000000000000		
XValues.Multiplier = 1.000000000000000000000000000000000000		
1.000000000000000000000000000000000000		
XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000000000000000000000	-	
YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000000000000000000000		
YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000 YValues.Order = loNone Left = 142 Top = 274 end end end end  C> object Chart3: TChart Left = 464 Top = 49 Width = 309 Height = 200 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Color = clSilver  LeftAxis.LabelsFont.Height = -13 LeftAxis.LabelsFont.Name = 'Arial' LeftAxis.LabelsFont.Style = [fsBold] LeftAxis.Title.Caption = 'Sales' LeftAxis.Title.Font.Charset = DEFAULT_CHARSET LeftAxis.Title.Font.Color = clGray LeftAxis.Title.Font.Height = -15 LeftAxis.Title.Font.Name = 'Arial' LeftAxis.Title.Font.Style = [fsBold] LeftAxis.Title.Font.Style = [fsBold] LeftAxis.Title.Font.Style = [fsBold] LeftAxis.Title.Font.Style = [fsBold]		
T Values.Nultiplier =  1.00000000000000000000  YValues.Order = loNone  Left = 142  Top = 274  end  end  end  ClBlue  LeftAxis.LabelsFont.Height = -13  LeftAxis.LabelsFont.Name =  'Arial'  LeftAxis.LabelsFont.Style =  [fsBold]  LeftAxis.Title.Caption = 'Sales'  LeftAxis.Title.Font.Charset =  DEFAULT CHARSET  LeftAxis.Title.Font.Color =  clGray  LeftAxis.Title.Font.Height = -15  LeftAxis.Title.Font.Height = -15  LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Style =  [fsBold]  LeftAxis.Title.Font.Style =  [fsBold]  LeftAxis.Title.Font.Style =  [fsBold]		
1.000000000000000000000000000000000000		
YValues.Order = loNone  Left = 142  Top = 274 end end end  Left = 464  Top = 49 Width = 309 Height = 200 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Color = clSilver  Left = 142 Left	-	
Left = 142		
Top = 274 end end end > object Chart3: TChart  Left = 464 Top = 49 Width = 309 Height = 200 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Color = clSilver  LeftAxis.LabelsFont.Style = [fsBold]  LeftAxis.Title.Caption = 'Sales'  LeftAxis.Title.Font.Charset = DEFAULT_CHARSET  LeftAxis.Title.Font.Color = clGray  LeftAxis.Title.Font.Height = -15  LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Style = [fsBold]  LeftAxis.Title.Font.Style = [fsBold]		
end end  Left = 464  Top = 49  Width = 309  Height = 200  BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Color = clSilver  [fsBold]  Left = 4sis.Title.Caption = 'Sales'  Left = 4sis.Title.Font.Charset = DEFAULT CHARSET  Left = 4sis.Title.Font.Color = clGray  Left = 4sis.Title.Font.Height = -15  Left = 4sis.Title.Font.Name = 'Arial'  Left = 4sis.Title.Font.Style = [fsBold]		
end  <> object Chart3: TChart  Left = 464  Top = 49  Width = 309  Height = 200  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  BackWall.Color = clSilver  LeftAxis.Title.Caption = 'Sales'  LeftAxis.Title.Font.Charset =  DEFAULT_CHARSET  LeftAxis.Title.Font.Color = clGray  LeftAxis.Title.Font.Height = -15  LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Style =  [fsBold]  Legend Visible = False	-	<u> </u>
<pre>Cond Cond Cond Cond Cond Cond Cond Cond</pre>		
Left = 464  Top = 49  Width = 309  Height = 200  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  BackWall.Color = clSilver  DEFAULT CHARSET  LeftAxis.Title.Font.Color = clGray  LeftAxis.Title.Font.Height = -15  LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Style = [fsBold]  Legend Visible = Felse	end	
Top = 49  Width = 309  Height = 200  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  BackWall.Color = clSilver  LeftAxis.Title.Font.Height = -15  LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Style = [fsBold]  Legond Visible = Felse	<> object Chart3: TChart	
Width = 309  Height = 200  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  BackWall.Color = clSilver  clGray  LeftAxis.Title.Font.Height = -15  LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Style = [fsBold]  Legend Visible = Felse	Left = 464	
Height = 200  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  BackWall.Color = clSilver  LeftAxis.Title.Font.Height = -15  LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Style = [fsBold]  Legend Visible = False	Top = 49	
BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Color = clSilver  LeftAxis.Title.Font.Name = 'Arial' LeftAxis.Title.Font.Style = [fsBold] Legend Visible = Felse	Width = 309	•
BackWall.Brush.Style = bsClear BackWall.Color = clSilver  LeftAxis.Title.Font.Style = [fsBold] Legend Visible = False		
BackWall.Color = clSilver  [fsBold]  Logond Visible = Folso	BackWall.Brush.Color = clWhite	
Logard Visible - False	BackWall.Brush.Style = bsClear	•
Foot.Alignment = taLeftJustify Legend.Visible = False	BackWall.Color = clSilver	
	Foot.Alignment = taLeftJustify	Legena. v isibie = r aise

DEEALILT CHARGET	Food Ford Charmet
DEFAULT_CHARSET  LeftAxis.Title.Font.Color =	Foot.Font.Charset = DEFAULT CHARSET
clGreen	Foot.Font.Color = clRed
LeftAxis.Title.Font.Height = -15	Foot.Font.Height = -12
LeftAxis.Title.Font.Name = 'Arial'	Foot.Font.Name = 'Arial'
LeftAxis.Title.Font.Style =	Foot.Font.Style = [fsItalic]
[fsBold, fsItalic]	Foot.Frame.Color = clScrollBar
Legend.Alignment = laTop	Foot.Text.Strings = (
Legend.Color = clGray	")
Legend.ColorWidth = 35	Gradient.Direction = gdLeftRight
Legend.Font.Charset =	Gradient.EndColor = clWhite
DEFAULT_CHARSET	Gradient.StartColor = clYellow
Legend.Font.Color = clWhite	LeftWall.Color = clSilver
Legend.Font.Height = -12	Title.Font.Charset =
Legend.Font.Name = 'Arial'	DEFAULT_CHARSET
Legend.Font.Style = [fsItalic]	Title.Font.Color = clRed
Legend.Frame.Color = clBlue	Title.Font.Height = -16
Legend.Frame.Width = 3	Title.Font.Name = 'Arial'
Legend.Frame.Visible = False	Title.Font.Style = [fsBold, fsItalic]
Legend.ShadowSize = 4	Title.Frame.Color = clScrollBar
Legend.TopPos = 34	Title.Text.Strings = (
RightAxis.Grid.Visible = False	'Area Series')
RightAxis.LabelsFont.Charset =	BackColor = clSilver
DEFAULT CHARSET	BottomAxis.DateTimeFormat =
RightAxis.LabelsFont.Color =	'd/mm'
clBlue	BottomAxis.LabelsAngle = 90
RightAxis.LabelsFont.Height = -	BottomAxis.LabelsFont.Charset =
13	DEFAULT CHARSET
RightAxis.LabelsFont.Name =	BottomAxis.LabelsFont.Color =
'Arial'	clTeal
RightAxis.LabelsFont.Style =	BottomAxis.LabelsFont.Height = -
[fsItalic]	12
RightAxis.Title.Caption = 'Red	BottomAxis.LabelsFont.Name =
Area'	'Arial'
RightAxis.Title.Font.Charset =	BottomAxis.LabelsFont.Style = []
DEFAULT CHARSET	BottomAxis.Title.Font.Charset =
RightAxis.Title.Font.Color =	DEFAULT CHARSET
clRed	BottomAxis.Title.Font.Color =
RightAxis.Title.Font.Height = -15	clGreen
RightAxis.Title.Font.Name =	BottomAxis.Title.Font.Height = -
'Arial'	16
RightAxis.Title.Font.Style =	BottomAxis.Title.Font.Name =
[fsBold, fsItalic]	'Arial'
Color = clWhite	
TabOrder = 4	BottomAxis.Title.Font.Style =
	[fsItalic] Chart3DParcent = 45
<> object South:	Chart3DPercent = 45
TAreaSeries  Marks Array Longth = 8	LeftAxis.Title.Caption = 'Green
Marks.ArrowLength = 8	Area'
Marks.Frame.Color = 8454143	LeftAxis.Title.Font.Charset =

Height = 173	Marks.Visible = False
BackWall.Brush.Color = clWhite	SeriesColor = clGreen
BackWall.Brush.Style = bsClear	DrawArea = True
BackWall.Color = 16777088	Pointer.InflateMargins = True
BackWall.Pen.Color = clBlue	Pointer.Style = psRectangle
BottomWall.Pen.Color = clBlue	Pointer.Visible = False
Foot.Font.Charset =	XValues.DateTime = True
DEFAULT_CHARSET	XValues.Name = 'X'
Foot.Font.Color = clYellow	XValues.Multiplier =
Foot.Font.Height = -11	1.000000000000000000
Foot.Font.Name = 'Arial'	XValues.Order = loAscending
Foot.Font.Style = []	YValues.DateTime = False
Foot.Frame.Color = clScrollBar	YValues.Name = 'Y'
Foot.Text.Strings = (	YValues.Multiplier =
'Last values')	1.0000000000000000000
Gradient.Direction =	YValues.Order = loNone
gdBottomTop	Left = 441
LeftWall.Color = clWhite	Top = 206
LeftWall.Pen.Color = clBlue	end
Title.Font.Charset =	<> object North:
DEFAULT_CHARSET	TAreaSeries
Title.Font.Color = clPurple	Marks.ArrowLength = 8
Title.Font.Height = -16	Marks.Frame.Color = 8454143
Title.Font.Name = 'Arial'	Marks.Visible = False
Title.Font.Style = [fsBold, fsItalic]	SeriesColor = clRed
Title.Frame.Color = clScrollBar	VertAxis = aRightAxis
Title.Text.Strings = (	DrawArea = True
'Point Series')	Pointer.InflateMargins = True
BackColor = 16777088	Pointer.Style = psRectangle
BottomAxis.Axis.Color = clBlue	Pointer.Visible = False
BottomAxis.Axis.Width = 1	XValues.DateTime = True
BottomAxis.DateTimeFormat =	XValues.Name = 'X'
'd/MM/yy'	XValues.Multiplier =
BottomAxis.Grid.Color = clBlue	1.000000000000000000
BottomAxis.LabelsFont.Charset =	XValues.Order = loAscending
DEFAULT_CHARSET	YValues.DateTime = False
BottomAxis.LabelsFont.Color =	YValues.Name = 'Y'
clGreen	YValues.Multiplier =
BottomAxis.LabelsFont.Height = -	1.0000000000000000000
12	YValues.Order = loNone
BottomAxis.LabelsFont.Name =	Left = 441
'Arial'	Top = 173
BottomAxis.LabelsFont.Style = []	end
BottomAxis.LabelsSeparation = 0	end
BottomAxis.MinorTickCount = 4	<> object Chart4: TChart
BottomAxis.TickLength = 3	Left = 466
<b>BottomAxis.Ticks.Color = clWhite</b>	Top = 256
BottomAxis.Title.Font.Charset =	Width = 311
DEFAULT_CHARSET	

# <u>AC.&P</u>

12
13
RightAxis.LabelsFont.Name =
'Arial'
RightAxis.LabelsFont.Style = []
RightAxis.Visible = False
TopAxis.Visible = False
Color = 8454143
TabOrder = 5
<> object Speaking:
<b>TPointSeries</b>
Marks.ArrowLength = 8
Marks.Frame.Color = 8454143
Marks.Visible = False
SeriesColor = clRed
VertAxis = aRightAxis
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = True
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000000000000000000000
YValues.Order = loNone
Left = 503
Top = 301
end
<> object Writing:
TPointSeries
Marks.ArrowLength = 8
Marks.Frame.Color = 8454143
Marks.Visible = False
SeriesColor = clGreen
Pointer.Brush.Color = clGreen
Pointer.InflateMargins = True
Pointer.Pen.Color = 64
Pointer.Style = psCircle
Pointer.Visible = True
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
1 values.Date i ilit – I aise

BottomAxis.Title.Font.Color =
clGreen
BottomAxis.Title.Font.Height = - 16
BottomAxis.Title.Font.Name =
'Arial'
BottomAxis.Title.Font.Style =
[fsItalic]
Chart3DPercent = 20
Frame.Color = clBlue
LeftAxis.Axis.Color = clBlue
LeftAxis.Axis.Width = 1
LeftAxis.Grid.Color = clBlue
LeftAxis.LabelsFont.Charset =
DEFAULT_CHARSET
LeftAxis.LabelsFont.Color =
clTeal
LeftAxis.LabelsFont.Height = -13
LeftAxis.LabelsFont.Name = 'Arial'
LeftAxis.LabelsFont.Style =
[fsItalic]
LeftAxis.Ticks.Color = clFuchsia
LeftAxis.Title.Font.Charset =
DEFAULT CHARSET
LeftAxis.Title.Font.Color =
clNavy
LeftAxis.Title.Font.Height = -15
LeftAxis.Title.Font.Name = 'Arial'
LeftAxis.Title.Font.Style =
[fsBold]
LeftAxis.Visible = False
Legend.Alignment = laBottom
Legend.ColorWidth = 50
Legend.Font.Charset =
DEFAULT_CHARSET
Legend.Font.Color = clPurple
Legend.Font.Height = -11
Legend.Font.Name = 'Arial'
Legend.Font.Style = []
LegendStyle =
lsLastValues
Legend.TopPos = 7
RightAxis.Grid.Color = clLime
RightAxis.LabelsFont.Charset =
DEFAULT_CHARSET
RightAxis.LabelsFont.Color = clYellow
RightAxis.LabelsFont.Height = -
raziitavistancisi alittigilt – -

Font.Color = clBlack
Font.Height = -16
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 0
OnClick = CheckBox1Click
end
<> object BitBtn1: TBitBtn
Left = 8
Top = 160
Width = 97
Height = 29
Caption = ' & Print all'
TabOrder = 1
OnClick = BitBtn1Click
Glyph.Data = {
NumGlyphs = 2
end
<> object RadioGroup1:
TRadioGroup
Left = 4
Top = 96
Width = 109
Height = 53
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -12
Font.Name = 'MS Sans Serif'
Font.Style = []
ItemIndex = 0
Items.Strings = (
'Proportional'
'More resolution')
ParentFont = False
TabOrder = 2
end
<> object CheckBox2:
TCheckBox
Left = 12
Top = 20
Width = 41
Height = 17
Caption = '3&D'
Checked = True
State = cbChecked TebOrdon = 3
TabOrder = 3

YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 501
Top = 267
end
<> object Reading:
<b>TPointSeries</b>
Marks.ArrowLength = 8
Marks.Frame.Color = 8454143
Marks.Visible = False
SeriesColor = clYellow
Pointer.Brush.Color = clYellow
Pointer.HorizSize = 5
Pointer.InflateMargins = True
Pointer.Style = psTriangle
Pointer.VertSize = 5
Pointer.Visible = True
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.0000000000000000000
YValues.Order = loNone
Left = 499
Top = 233
end
end
<> object Panel1: TPanel
Left = -1
Top = 40
Width = 121
Height = 417
TabOrder = 1
<> object CheckBox1:
TCheckBox
Left = 12
Top = 52
Width = 105
Height = 17
Caption = '&Animate !!!'
Font.Charset =
DEFAULT_CHARSET

ParentFont = False
end
end
<> object Timer1: TTimer
Enabled = False
Interval = 1
OnTimer = Timer1Timer
Left = 102
Top = 254
end
<> object SaveDialog1:
TSaveDialog
DefaultExt = 'BMP'
FileName = 'TEEDEMO.BMP'
Filter = 'Bitmap Files *.bmp'
Options = [ofOverwritePrompt,
ofHideReadOnly]
Title = 'TeeChart DEMO Save to
Bitmap'
Left = 162
Top = 444
end
End
\\\\\\\
<> object BitmapForm:
TBitmapForm
Left = 183
Top = 118 Width = 675
Height = 421
Caption = 'TeeChart Background
Image demo'
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Chart1: TChart
Left = 145
Top = 0
Width = 522
Height = 387
BackImage.Data = {
07544269746D6170E62B0000424DE
62B0000000000007600000028000000

1
OnClick = CheckBox2Click
end
<> object BitBtn2: TBitBtn
Left = 8
Top = 264
Width = 97
Height = 29
Caption = ' &Save all '
TabOrder = 4
OnClick = BitBtn2Click
Glyph.Data = {
•
NumGlyphs = 2
end
<> object BitBtn3: TBitBtn
Left = 8
Top = 368
Width = 100
Height = 37
Caption = ' Close'
TabOrder = 5
Kind = bkClose
end
end
<> object Panel2: TPanel
Left = 0
Top = 0
Width = 777
Height = 40
Align = alTop
TabOrder = 2
<> object Label1: TLabel
Left = 12
Top = 12
Width = 471
Height = 16
HelpType = htKeyword
Alignment = taCenter
Caption =
'Left drag (Up/Down) to Zoom.
Right drag to Scroll. Invert Zoom r'
+
'ectangle to reset.'
Font.Charset =
ANSI CHARSET
Font.Color = clNavy
Font.Height = -13
Font.Name = 'Arial'
Font.Style = [fsItalic]
1 onto yie [istane]

Left = 0
Top = 0
Width = 145
Height = 387
Align = alLeft
TabOrder = 1
<> object RadioGroup1:
TRadioGroup
Left = 22
Top = 12
Width = 91
Height = 85
Caption = 'Bitmap Mode:'
ItemIndex = 1
Items.Strings = (
'Stretch'
'Tile'
'Center')
TabOrder = 0
OnClick = RadioGroup1Click
end
<> object BitBtn1: TBitBtn
Left = 12
Top = 132
Width = 125
Height = 25
Caption = '&Open Bitmap'
TabOrder = 1
OnClick = BitBtn1Click
Glyph.Data = {
NumGlyphs = 2
end
<> object BitBtn3: TBitBtn
Left = 28
Top = 348
Width = 85
Height = 37
TabOrder = 2
Kind = bkClose
end
<> object CheckBox1: TCheckBox
Left = 36
Top = 104 Width = 65
Height = 17
Charlest - True
Checked = True

9900
BackImageInside = True
BackImageMode = pbmTile
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
Foot.Font.Charset =
DEFAULT CHARSET
Foot.Font.Color = clRed
Foot.Font.Height = -12
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsItalic]
Foot.Frame.Color = clScrollBar
Title.Brush.Color = clBlack
Title.Color = clBlack
Title.Font.Charset =
DEFAULT CHARSET
Title.Font.Color = clWhite
Title.Font.Height = -21
Title.Font.Name = 'Arial'
Title.Font.Style = [fsBold, fsItalic]
Title.Frame.Color = clScrollBar
Title.Text.Strings = (
'TeeChart Panel Bitmap
Example')
Chart3DPercent = 45
Legend.TopPos = 20
Align = alClient
BevelInner = bvRaised
BorderWidth = 28
TabOrder = 0
<> object Series1:
TBarSeries
Marks.ArrowLength = 20
Marks.Visible = True
SeriesColor = clRed
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Bar'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
end
end
<> object Panel1: TPanel

ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 153
Height = 355
Align = alLeft
TabOrder = 0
<> object CheckBox1:
TCheckBox
Left = 20
Top = 13
Width = 125
Height = 17
Caption = 'Animate !!!'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -19
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 0
OnClick = CheckBox1Click
end
<> object BitBtn3: TBitBtn
Left = 32
Top = 309
Width = 89
Height = 33
TabOrder = 1
Kind = bkClose
end
<> object Memo1: TMemo
Left = 16
Top = 48
Width = 121
Height = 137
Lines.Strings = (
'Each Arrow is '
'represented as a Series '
'point with Starting and '
'Ending coordinates.'

State = cbChecked
TabOrder = 3
OnClick = CheckBox1Click
end
<> object Button1: TButton
Left = 12
Top = 172
Width = 125
Height = 25
Caption = '&Clear'
TabOrder = 4
OnClick = Button1Click
end
<> object Memo1: TMemo
Left = 13
Top = 216
Width = 121
Height = 105
Lines.Strings = (
'Images can be used as '
'backgrounds in '
'centered,'
'tiled or stretched mode.')
TabOrder = 5
end
end
<> object OpenDialog1:
TOpenDialog
DefaultExt = 'BMP'
Filter = 'Bitmap Files *.BMP'
Options = [ofReadOnly,
ofPathMustExist, ofFileMustExist]
Title = 'Select a Bitmap File'
Left = 100
Top = 43
end
End
111111111111111111111111111111111111111
<> object ArrowsForm:
TArrowsForm
Left = 188
Top = 111
Width = 600
Height = 389
Caption = 'TeeChart Arrow Series
Example'
Color = clBtnFace

Pointer.HorizSize = 32
Pointer.InflateMargins = False
Pointer.Style = psRectangle
Pointer.VertSize = 24
Pointer.Visible = True
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.0000000000000000000000000000000000000
YValues.Order = loNone
EndXValues.DateTime = False
EndX Values. Date 1 me - Paise  EndXValues. Name = 'EndX'
EndXValues.Multiplier = 1.00000000000000000000000000000000000
EndXValues.Order = loNone
EndYValues.DateTime = False
EndYValues.Name = 'EndY'
EndYValues.Multiplier =
1.00000000000000000
EndYValues.Order = loNone
StartXValues.DateTime = True
StartXValues.Name = 'X'
StartXValues.Multiplier =
1.00000000000000000
StartXValues.Order =
loAscending
StartYValues.DateTime = False
StartYValues.Name = 'Y'
StartYValues.Multiplier =
1.000000000000000000
StartYValues.Order = loNone
Left = 104
Top = 195
end
end
<> object Timer1: TTimer
Enabled = False
Interval = 1
OnTimer = Timer1Timer
Left = 62
Top = 195
end
End

"
'This demo changes '
'arrow positions randomly '
'using a'
'TTimer component.')
TabOrder = 2
end
end
<> object Chart1: TChart
Left = 153
Top = 0
Width = 439
Height = 355
AnimatedZoom = True
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Color = 8454143
BottomWall.Color = 8454016
Foot.Font.Charset =
DEFAULT_CHARSET
Foot.Font.Color = clRed
Foot.Font.Height = -12
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsItalic]
Foot.Frame.Color = clScrollBar
LeftWall.Color = 16777088
Title.Frame.Color = clScrollBar
Title.Text.Strings = (
")
Title.Visible = False
BackColor = 8454143
Chart3DPercent = 50
Legend.Visible = False
RightAxis.Grid.Visible = False
RightAxis.MinorTickLength = 3
RightAxis.TickLength = 5
Align = alClient
TabOrder = 1
<> object ArrowSeries1:
TArrowSeries
ColorEachPoint = True
HorizAxis = aTopAxis
Marks.ArrowLength = 8
Marks.Frame.Visible = False
Marks.Transparent = True
Marks.Visible = False
SeriesColor = clRed
VertAxis = aRightAxis

'maStacked'
'maStacked100')
TabOrder = 0
OnClick = RadioGroup1Click
end
<> object BitBtn1: TBitBtn
Left = 18
Top = 350
Width = 89
Height = 33
TabOrder = 1
Kind = bkClose
end
<> object CheckBox1:
TCheckBox
Left = 14
Top = 34
Width = 97
Height = 17
Caption = '3D'
Checked = True
State = cbChecked
TabOrder = 2
OnClick = CheckBox1Click
end
<> object CheckBox2:
TCheckBox
Left = 14
Top = 9
Width = 97
Height = 17
Caption = '&Animate !!!'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -16
Font.Name = 'Arial'
Font.Name - Ariai Font.Style = [fsBold]
ParentFont = False
TabOrder = 3
OnClick = CheckBox2Click
end
end
<> object Chart1: TChart
Left = 145
Top = 0
Width = 445
Height = 399

[
<> object AreasForm:
TAreasForm
Left = 192
Top = 110
Width = 598
Height = 433
ActiveControl = Chart1
Caption = 'Area Series Example'
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 145
Height = 399
Align = alLeft
TabOrder = 0
<> object Label1: TLabel
Left = 14
Top = 196
Width = 62
Height = 22
HelpType = htKeyword
Caption = 'Label1'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -19
Font.Name = 'Arial'
Font.Style = [fsBold, fsItalic]
ParentFont = False
end
<> object RadioGroup1:
TRadioGroup
Left = 12
Top = 59
Width = 117
Height = 122
Caption = 'Area Series Mode:'
ItemIndex = 0
Items.Strings = ( 'maNone'
marvone

end
<> object AreaSeries2:
TAreaSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clGreen
DrawArea = True
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 227
Top = 36
end
<> object AreaSeries3:
TAreaSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clYellow
DrawArea = True
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 265
Top = 34
end
, a
end
<pre>end &lt;&gt; object Timer1: TTimer Enabled = False</pre>

	BackWall.Brush.Color = clWhite
	BackWall.Brush.Style = bsClear
	Foot.Font.Charset =
	DEFAULT CHARSET
	Foot.Font.Color = clRed
	Foot.Font.Height = -12
	Foot.Font.Name = 'Arial'
	Foot.Font.Style = [fsItalic]
	Title.Alignment = taLeftJustify
	Title.Font.Charset =
	DEFAULT CHARSET
1	Title.Font.Color = clBlue
	Title.Font.Height = -13
	Title.Font.Name = 'Arial'
	Title.Font.Style = [fsBold]
	• • •
	Title.Text.Strings = (
	'Area Series Example.'
	'Move mouse over Areas.')
	BottomAxis.LabelsAngle = 90
	Chart3DPercent = 50
	Legend.Alignment = laBottom
	Align = alClient
	TabOrder = 1
	OnMouseMove =
	Chart1MouseMove
	<> object AreaSeries1:
	TAreaSeries
	Marks.ArrowLength = 8
	Marks.Visible = False
	SeriesColor = clRed
	AreaLinesPen.Color = clAqua
	AreaLinesPen.Style = psDot
	DrawArea = True
	Pointer.InflateMargins = True
1	Pointer.Style = psRectangle
	Pointer.Visible = False
1	XValues.DateTime = True
1	XValues.Name = 'X'
1	XValues.Nultiplier =
1	1.000000000000000000000000000000000000
	XValues.Order = loAscending
	YValues Name = IV!
-	YValues.Name = 'Y'
4	YValues.Multiplier =
4	1.000000000000000000
4	YValues.Order = loNone
4	Left = 189
	Top = 36

end
<> object Label2: TLabel
Left = 168
Top = 353
Width = 300
Height = 13
HelpType = htKeyword
Caption = '© Copyright 1996-2001
by David Berneda. All Rights
Reserved.'
Color = clNavy
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWhite
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentColor = False
ParentFont = False
end
<> object Panel1: TPanel
Left = 0
Top = 376
Width = 632
Height = 30
Align = alBottom
Alignment = taLeftJustify
TabOrder = 0
<> object Label11: TLabel
Left = 12
Top = 49
Width = 3
Height = 13
HelpType = htKeyword
end
<> object Label13: TLabel
Left = 12
Top = 6
Width = 129
Height = 19
HelpType = htKeyword
Caption = 'Steema Software' Font.Charset =
DEFAULT CHARSET
Font Height = 16
Font Name = !Aria!!
Font.Name = 'Arial'
Font.Style = [fsBold]

Interval = 1
OnTimer = Timer1Timer
Left = 88
Top = 91
end
End
<> object TeeMainForm:
TTeeMainForm
Left = 225
Top = 188
ActiveControl = Button2
BorderIcons = [biSystemMenu]
BorderStyle = bsDialog
Caption = 'TeeChart Standard
version 4.0 - Demo'
ClientHeight = 406
ClientWidth = 632
Color = clNavy
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Image1: TImage
Left = 484
Top = 26
Width = 141
Height = 45
HelpType = htKeyword
AutoSize = True
Picture.Data = {
07544269746D6170861D0000424D86
1D00000000000036040000280000008
D00
00002D000000010008000000000501
900000000000000000000000000000000000000
01
OnClick = Label10Click

end
<> object Button1: TButton
Left = 536
<b>Top</b> = 4
Width = 89
Height = 23
Caption = 'Close'
ModalResult = 1
TabOrder = 0
OnClick = Button1Click
end
end
<> object TPanel
Left = 476
Top = 102
Width = 157
Height = 251
BevelOuter = bvNone
BorderWidth = 1
Color = clNavy
TabOrder = 1
<> object Button2: TButton
Left = 8
Top = -1
Width = 141
Height = 41
Caption = 'Show me the Demo !'
Default = True
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -12
Font.Name = 'Arial'
Font.Style = [fsBold] ParentFont = False
TabOrder = 0
OnClick = Button2Click
end
<> object Button3: TButton
Left = 8
Top = 85
Width = 141
Height = 29
Caption = 'Features'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -12
I OHIGHTOIGHT 12

ParentFont = False
end
<> object Label15: TLabel
Left = 212
Top = 9
<b>Width = 27</b>
Height = 13
HelpType = htKeyword
Caption = 'email:'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
<> object Label16: TLabel
Left = 244
Top = 9
Width = 85
Height = 13
HelpType = htKeyword
Caption = 'info@steema.com'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
<> object Label3: TLabel
Left = 375
Top = 6
Width = 119
Height = 16
Cursor = crHandPoint
HelpType = htKeyword
Caption = 'www.steema.com'
Font.Charset =
DEFAULT CHARSET
_
Font.Color = clBlue
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold, fsUnderline]
ParentFont = False
OnClick = Label3Click

Title.Visible = False
AxisVisible = False
ClipPoints = False
Frame.Visible = False
Legend.Color = clSilver
Legend.Visible = False
View3DWalls = False
BevelWidth = 2
Color = clSilver TabOrder = 4
OnMouseDown =
Chart2MouseDown
OnMouseUp = Chart2MouseUp
<> object Label12: TLabel
Left = 25
Top = 16
<b>Width = 97</b>
Height = 32
HelpType = htKeyword
Caption = 'About'
Font.Charset =
DEFAULT CHARSET
Font.Color = clRed
Font.Height = -29
Font.Name = 'Times New
Roman'
Font.Style = [fsBold]
ParentFont = False
Transparent = True OnClick = Label10Click
OnMouseDown =
Chart2MouseDown
end
<> object FastLineSeries1:
TFastLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clAqua
LinePen.Color = clAqua
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.00000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier = 1.00000000000000000000000000000000000
YValues.Order = loNone

77 (37 14 1 11
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 1
OnClick = Button3Click
end
<> object Button4: TButton
Left = 8
Top = 122
Width = 141
Height = 29
Caption = 'Pro Specifications'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -12
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 2
OnClick = Button4Click
end
<> object Button6: TButton
Left = 8
Top = 48
Width = 141
Height = 29
Caption = 'Chart Styles'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -12
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 3
OnClick = Button6Click
end
<> object Chart2: TChart
Left = 8
Top = 168
Width = 141
Height = 73
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Pen.Visible = False
Title.Text.Strings = (
'TChart')

Title.Text.Strings = (  '')  AxisVisible = False Chart3DPercent = 50 ClipPoints = False Frame.Visible = False Legend.Visible = False View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DOptions.Zoom = 92 View3DWalls = False BevelOuter = bvNone Color = clAqua TabOrder = 1 <> object PieSeries6: TPieSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed CustomYRadius = 25 ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False PieValues.Name = 'Pie'
Chart3DPercent = 50 ClipPoints = False Frame.Visible = False Legend.Visible = False View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DOptions.Zoom = 92 View3DWalls = False BevelOuter = bvNone Color = clAqua TabOrder = 1 <> object PieSeries6: TPieSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed CustomYRadius = 25 ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
Chart3DPercent = 50 ClipPoints = False Frame.Visible = False Legend.Visible = False View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DOptions.Zoom = 92 View3DWalls = False BevelOuter = bvNone Color = clAqua TabOrder = 1 <> object PieSeries6: TPieSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed CustomYRadius = 25 ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
ClipPoints = False Frame.Visible = False Legend.Visible = False View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DOptions.Zoom = 92 View3DWalls = False BevelOuter = bvNone Color = clAqua TabOrder = 1 <> object PieSeries6: TPieSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed CustomYRadius = 25 ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
Frame.Visible = False Legend.Visible = False View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DOptions.Zoom = 92 View3DWalls = False BevelOuter = bvNone Color = clAqua TabOrder = 1 <> object PieSeries6: TPieSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed CustomYRadius = 25 ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
Legend.Visible = False View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DOptions.Zoom = 92 View3DWalls = False BevelOuter = bvNone Color = clAqua TabOrder = 1 <> object PieSeries6: TPieSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed CustomYRadius = 25 ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DOptions.Zoom = 92 View3DWalls = False BevelOuter = bvNone Color = clAqua TabOrder = 1 <> object PieSeries6: TPieSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed CustomYRadius = 25 ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DOptions.Zoom = 92 View3DWalls = False BevelOuter = bvNone Color = clAqua TabOrder = 1 <> object PieSeries6: TPieSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed CustomYRadius = 25 ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DOptions.Zoom = 92 View3DWalls = False BevelOuter = bvNone Color = clAqua TabOrder = 1 <> object PieSeries6: TPieSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed CustomYRadius = 25 ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DOptions.Zoom = 92 View3DWalls = False BevelOuter = bvNone Color = clAqua TabOrder = 1 <> object PieSeries6: TPieSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed CustomYRadius = 25 ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
View3DOptions.Rotation = 360 View3DOptions.Zoom = 92 View3DWalls = False BevelOuter = bvNone Color = clAqua TabOrder = 1 <> object PieSeries6: TPieSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed CustomYRadius = 25 ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
View3DOptions.Zoom = 92 View3DWalls = False BevelOuter = bvNone Color = clAqua TabOrder = 1 <> object PieSeries6: TPieSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed CustomYRadius = 25 ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
View3DOptions.Zoom = 92 View3DWalls = False BevelOuter = bvNone Color = clAqua TabOrder = 1 <> object PieSeries6: TPieSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed CustomYRadius = 25 ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
View3DWalls = False BevelOuter = bvNone Color = clAqua TabOrder = 1 <> object PieSeries6: TPieSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed CustomYRadius = 25 ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
Color = clAqua TabOrder = 1 <> object PieSeries6: TPieSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed CustomYRadius = 25 ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
Color = clAqua TabOrder = 1 <> object PieSeries6: TPieSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed CustomYRadius = 25 ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
TabOrder = 1 <> object PieSeries6: TPieSeries  Marks.ArrowLength = 8  Marks.Visible = False SeriesColor = clRed CustomYRadius = 25  ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
<> object PieSeries6: TPieSeries  Marks.ArrowLength = 8  Marks.Visible = False SeriesColor = clRed CustomYRadius = 25  ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
TPieSeries  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clRed  CustomYRadius = 25  ExplodeBiggest = 25  OtherSlice.Text = 'Other'  PieValues.DateTime = False
Marks.ArrowLength = 8  Marks.Visible = False SeriesColor = clRed CustomYRadius = 25  ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
Marks.Visible = False SeriesColor = clRed CustomYRadius = 25 ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
SeriesColor = clRed CustomYRadius = 25 ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
CustomYRadius = 25 ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
ExplodeBiggest = 25 OtherSlice.Text = 'Other' PieValues.DateTime = False
OtherSlice.Text = 'Other' PieValues.DateTime = False
PieValues.DateTime = False
1 ic v alucs. value – 1 ic
PieValues.Multiplier =
1.000000000000000000000000000000000000
PieValues.Order = loNone
end
end
<> object Chart8: TChart
Left = 3
Top = 163
Width = 97
Height = 66
AllowPanning = pmNone
AllowZoom = False
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Pen.Visible = False
MarginBottom = 10
MarginLeft = 5
MarginRight = 5
MarginTop = 0
Title.Font.Charset =
DEFAULT_CHARSET

end
<> object FastLineSeries2:
TFastLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clYellow
LinePen.Color = clYellow
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.00000000000000000000
YValues.Order = loNone
end
end
end
<> object Panel2: TPanel
Left = 9
Top = 102
Width = 465
Height = 241
BevelWidth = 2
Color = clAqua
TabOrder = 2
<> object Chart7: TChart
Left = 3
<b>Top</b> = 7
Width = 97
Height = 66
AllowPanning = pmNone
AllowZoom = False
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Pen.Visible = False
MarginBottom = 10
MarginLeft = 5
MarginRight = 5
MarginTop = 0
Title.Font.Charset =
DEFAULT CHARSET
Title.Font.Color = clBlack
Title Font Name = 'A wiel'
Title.Font.Name = 'Arial'
Title.Font.Style = [fsBold]

MarginRight = 5
MarginTop = 0
Title.Font.Charset =
DEFAULT_CHARSET
Title.Font.Color = clBlack
Title.Font.Height = -17
Title.Font.Name = 'Arial'
Title.Font.Style = [fsBold]
Title.Text.Strings = (
'')
AxisVisible = False
Chart3DPercent = 100
ClipPoints = False
Frame.Visible = False
Legend. Visible = False
View3DOptions.Elevation = 315
View3DOptions.Orthogonal =
False
View3DOptions.Perspective = 0
View3DOptions.Rotation = 360
View3DOptions.Zoom = 92
View3DWalls = False
BevelOuter = bvNone
Color = clAqua
TabOrder = 4
<> object PieSeries9:
TPieSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clRed
CustomYRadius = 25
ExplodeBiggest = 40
OtherSlice.Text = 'Other'
PieValues DateTime = False
PieValues.Name = 'Pie'
PieValues.Multiplier =
1.00000000000000000
PieValues.Order = loNone
end
end
<> object Chart9: TChart
Left = 206
Top = 4
Width = 255
Height = 221
AllowPanning = pmNone
AllowZoom = False
BackWall.Brush.Color = clWhite

Title.Font.Color = clBlack
Title.Font.Height = -17
Title.Font.Name = 'Arial'
Title.Font.Style = [fsBold]
Title.Text.Strings = (
"')
AxisVisible = False
Chart3DPercent = 95
ClipPoints = False
Frame.Visible = False
Legend.Visible = False
View3DOptions.Elevation = 315
View3DOptions.Orthogonal =
False
View3DOptions.Perspective = 0
View3DOptions.Rotation = 360
View3DOptions.Zoom = 87
View3DWalls = False
BevelOuter = bvNone
Color = clAqua
TabOrder = 2
<> object PieSeries7:
TPieSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clRed
CustomYRadius = 25
ExplodeBiggest = 20
OtherSlice.Text = 'Other'
PieValues.DateTime = False
PieValues.Name = 'Pie'
PieValues.Multiplier =
1.000000000000000000000000000000000000
PieValues.Order = loNone
end
end
<> object Chart10: TChart
Left = 105
Top = 139
Width = 97
Height = 66
AllowPanning = pmNone
AllowZoom = False
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Pen.Visible = False
MarginBottom = 10
MarginLeft = 5

# <u>AC.&P</u>

OtherSlice.Text = 'Other'
PiePen.Visible = False
PieValues.DateTime = False
PieValues.Name = 'Pie'
PieValues.Multiplier =
1.00000000000000000
PieValues.Order = loNone
DataSources = (
'Series1'
'PieSeries6'
'PieSeries7'
'PieSeries9')
<> object
PieSeries8TAddTeeFunction:
TAddTeeFunction
end
end
end
<> object Chart1: TChart
Left = 105
Top = 31
Width = 97
Height = 66
AllowPanning = pmNone
AllowZoom = False
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Pen.Visible = False
MarginBottom = 10
MarginLeft = 5
MarginRight = 5
MarginTop = 0
Title.Font.Charset =
DEFAULT_CHARSET
Title.Font.Color = clBlack
Title.Font.Height = -17
Title.Font.Name = 'Arial'
Title.Font.Style = [fsBold]
Title.Text.Strings = (
")
AxisVisible = False
Chart3DPercent = 50
ClipPoints = False
Frame.Visible = False
Legend.Visible = False
View3DOptions.Elevation = 315
View3DOptions.Orthogonal =
False

BackWall.Brush.Style = bsClear
BackWall.Pen.Visible = False
MarginBottom = 10
MarginLeft = 5
MarginRight = 5
MarginTop = 0
Title.Font.Charset =
DEFAULT CHARSET
Title.Font.Color = clBlack
Title.Font.Height = -24
Title.Font.Name = 'Arial'
Title.Font.Style = [fsBold]
Title.Text.Strings = (
")
Title.Visible = False
AxisVisible = False
Chart3DPercent = 70
ClipPoints = False
Frame.Visible = False
Legend.TextStyle = ltsPlain
Legend. Visible = False
View3DOptions.Elevation = 315
View3DOptions.Orthogonal =
False
View3DOptions.Perspective = 0
View3DOptions.Rotation = 360
View3DWalls = False
BevelOuter = bvNone
Color = clAqua
TabOrder = 3
<> object PieSeries8:
TPieSeries
Marks.Arrow.Color = clBlack
Marks.ArrowLength = 1 Marks.Font.Charset =
Marks.ront.Cnarset = DEFAULT CHARSET
Marks.Font.Color = clNavy
Marks.Font.Height = -11
Marks.Font.Name = 'Arial'
Marks.Font.Style = []
Marks.Frame.Visible = False
Marks.Transparent = True
Marks.Visible = True
DataSource = Series1
PercentFormat = '##0.00 %'
SeriesColor = clRed
CustomYRadius = 35
ExplodeBiggest = 40

Alignment = taCenter
AutoSize = False
Caption = 'The 100% Native
Charting VCL / CLX Library'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -19
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
end
<> object Label9: TLabel
Left = 8
Top = 44
Width = 449
Height = 19
HelpType = htKeyword
Alignment = taCenter
AutoSize = False
Caption = 'For Borland Delphi,
Kylix and C++ Builder'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clNavy
Font.Height = -16
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
end
end
<> object Timer1: TTimer
Interval = 1
OnTimer = Timer1Timer
Left = 408
Top = 283
end
<> object Timer2: TTimer
Interval = 200
OnTimer = Timer2Timer
Left = 356
Top = 281
end
End
<> object MetafileForm:
TMetafileForm

View3DOptions.Perspective = 0
View3DOptions.Rotation = 360
View3DWalls = False
BevelOuter = bvNone
Color = clAqua
TabOrder = 0
<> object Series1:
TPieSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clRed
CustomYRadius = 25
ExplodeBiggest = 25
OtherSlice.Text = 'Other'
PieValues.DateTime = False
PieValues.Name = 'Pie'
PieValues.Multiplier =
1.000000000000000000
PieValues.Order = loNone
end
end
end
<> object TPanel
Left = 8
Top = 359
Width = 156
Height = 1
TabOrder = 3
end
<> object TPanel
Left = 475
Top = 360
Width = 153
Height = 1
TabOrder = 4
end
<> object Panel3: TPanel
Left = 8
Top = 11
Width = 465
Height = 81
TabOrder = 5
<> object Label1: TLabel
Left = 8
Top = 15
Width = 449
Height = 22
HelpType = htKeyword
Theip Type - neixey word

Align = alTop
TabOrder = 0
<> object BarSeries1:
TBarSeries
ColorEachPoint = True
Marks.ArrowLength = 20
Marks.Visible = True
SeriesColor = clRed
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000000000000000000000
YValues.Order = loNone
Left = 15
Top = 52 end
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 546
Height = 41
Align = alTop
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'Arial'
Font.Style = []
ParentFont = False
TabOrder = 1
<> object BitBtn2: TBitBtn
Left = 12
Top = 4
Width = 149
Height = 33
Caption = '&Save to *.WMF'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
rondstyte – []

<u> </u>
Left = 199
Top = 116
Width = 554
Height = 443
Caption = 'TeeChart Metafile
Format Demo'
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Image1: TImage
Left = 0
Top = 258
Width = 546
Height = 151
HelpType = htKeyword
Align = alClient
Stretch = True
end
<> object Chart1: TChart
Left = 0
Top = 41
Width = 546
Height = 176
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
Foot.Font.Charset =
DEFAULT_CHARSET
Foot.Font.Color = clRed
Foot.Font.Height = -12
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsItalic]
Foot.Frame.Color = clScrollBar
Title.Frame.Color = clWhite
Title.Text.Strings = (
'TChart Component')
BottomAxis.DateTimeFormat =
'MM/dd/yy'
BottomAxis.Grid.Color =
clScrollBar
Chart3DPercent = 35
LeftAxis.Grid.Color = clScrollBar
RightAxis.Grid.Color =
clScrollBar
TopAxis.Grid.Color = clScrollBar

End	_
	end
C> object PagesForm: TPagesForm  Left = 203  Top = 106  Width = 608  Height = 413  Caption = 'TeeChart Sample. How to divide Charts in several Pages.'  Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style =    OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96  TextHeight = 13  <> object Chart1: TChart Left = 194  Top = 0  Width = 406 Height = 379 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Text.Strings = (  'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	End
C> object PagesForm: TPagesForm  Left = 203  Top = 106  Width = 608  Height = 413  Caption = 'TeeChart Sample. How to divide Charts in several Pages.'  Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style =    OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96  TextHeight = 13  <> object Chart1: TChart Left = 194  Top = 0  Width = 406 Height = 379 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Text.Strings = (  'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	
TPagesForm  Left = 203  Top = 106  Width = 608  Height = 413  Caption = 'TeeChart Sample. How to divide Charts in several Pages.'  Color = clBtnFace Font.Charset =  DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11  Font.Name = 'MS Sans Serif' Font.Style = []  OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96  TextHeight = 13  <> object Chart1: TChart  Left = 194  Top = 0  Width = 406  Height = 379  BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'TChart Component')  BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  RightAxis.Grid.Color = clScrollBar  OnPageChange = Chart1PageChange  Align = alClient  TabOrder = 0  <> object LineSeries1:	
Left = 203  Top = 106  Width = 608  Height = 413  Caption = 'TeeChart Sample. How to divide Charts in several Pages.'  Color = clBtnFace Font.Charset =  DEFAULT_CHARSET  Font.Color = clBlack Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style =     OldCreateOrder = True  Position = poScreenCenter  OnCreate = FormCreate  PixelsPerInch = 96  TextHeight = 13  <> object Chart1: TChart  Left = 194  Top = 0  Width = 406  Height = 379  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  Foot.Frame.Color = clScrollBar  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'TChart Component')  BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  RightAxis.Grid.Color = clScrollBar  OnPageChange = Chart1PageChange  Align = alClient  TabOrder = 0  <> object LineSeries1:	
Top = 106  Width = 608  Height = 413  Caption = 'TeeChart Sample. How to divide Charts in several Pages.'  Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 194 Top = 0 Width = 406 Height = 379 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = (     'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend,Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	
Width = 608 Height = 413 Caption = 'TeeChart Sample. How to divide Charts in several Pages.' Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 194 Top = 0 Width = 406 Height = 379 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = (     'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	
Height = 413  Caption = 'TeeChart Sample. How to divide Charts in several Pages.'  Color = clBtnFace  Font.Charset = DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  OldCreateOrder = True  Position = poScreenCenter  OnCreate = FormCreate  PixelsPerInch = 96  TextHeight = 13  <> object Chart1: TChart  Left = 194  Top = 0  Width = 406  Height = 379  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  Foot.Frame.Color = clScrollBar  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'TChart Component')  BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  RightAxis.Grid.Color = clScrollBar  ScaleLastPage = False  TopAxis.Grid.Color = clScrollBar  OnPageChange = Chart1PageChange  Align = alClient  TabOrder = 0  <> object LineSeries1:	
Caption = 'TeeChart Sample. How to divide Charts in several Pages.'  Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style =    OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96  TextHeight = 13 <> object Chart1: TChart Left = 194 Top = 0 Width = 406 Height = 379 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	
to divide Charts in several Pages.'  Color = clBtnFace  Font.Charset =  DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  OldCreateOrder = True  Position = poScreenCenter  OnCreate = FormCreate  PixelsPerInch = 96  TextHeight = 13  <> object Chart1: TChart  Left = 194  Top = 0  Width = 406  Height = 379  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  Foot.Frame.Color = clScrollBar  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'TChart Component')  BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  RightAxis.Grid.Color = clScrollBar  ScaleLastPage = False  TopAxis.Grid.Color = clScrollBar  OnPageChange =  Chart1PageChange  Align = alClient  TabOrder = 0  <> object LineSeries1:	
Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 194 Top = 0 Width = 406 Height = 379 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange Align = alClient TabOrder = 0 <> object LineSeries1:	
Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 194 Top = 0 Width = 406 Height = 379 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	
DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 194 Top = 0 Width = 406 Height = 379 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	
Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 194 Top = 0 Width = 406 Height = 379 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	_ 0 0 0 0 0 0 0 0
Font.Name = 'MS Sans Serif'  Font.Style = []  OldCreateOrder = True  Position = poScreenCenter  OnCreate = FormCreate  PixelsPerInch = 96  TextHeight = 13  <> object Chart1: TChart  Left = 194  Top = 0  Width = 406  Height = 379  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  Foot.Frame.Color = clScrollBar  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'TChart Component')  BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  RightAxis.Grid.Color = clScrollBar  ScaleLastPage = False  TopAxis.Grid.Color = clScrollBar  OnPageChange =  Chart1PageChange  Align = alClient  TabOrder = 0  <> object LineSeries1:	
Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 194 Top = 0 Width = 406 Height = 379 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	
Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 194 Top = 0 Width = 406 Height = 379 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	Ÿ
OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 194 Top = 0 Width = 406 Height = 379 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	
Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 194 Top = 0 Width = 406 Height = 379 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	
OnCreate = FormCreate PixelsPerInch = 96  TextHeight = 13  <> object Chart1: TChart  Left = 194  Top = 0  Width = 406  Height = 379  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  Foot.Frame.Color = clScrollBar  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'TChart Component')  BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  RightAxis.Grid.Color = clScrollBar  ScaleLastPage = False  TopAxis.Grid.Color = clScrollBar  OnPageChange =  Chart1PageChange  Align = alClient  TabOrder = 0  <> object LineSeries1:	OldCreateOrder = True
PixelsPerInch = 96  TextHeight = 13  <> object Chart1: TChart  Left = 194  Top = 0  Width = 406  Height = 379  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  Foot.Frame.Color = clScrollBar  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'TChart Component')  BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  RightAxis.Grid.Color = clScrollBar  ScaleLastPage = False  TopAxis.Grid.Color = clScrollBar  OnPageChange =  Chart1PageChange  Align = alClient  TabOrder = 0  <> object LineSeries1:	Position = poScreenCenter
TextHeight = 13  <> object Chart1: TChart  Left = 194  Top = 0  Width = 406  Height = 379  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  Foot.Frame.Color = clScrollBar  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'TChart Component')  BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  RightAxis.Grid.Color = clScrollBar  ScaleLastPage = False  TopAxis.Grid.Color = clScrollBar  OnPageChange =  Chart1PageChange  Align = alClient  TabOrder = 0  <> object LineSeries1:	OnCreate = FormCreate
<> object Chart1: TChart  Left = 194  Top = 0  Width = 406  Height = 379  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  Foot.Frame.Color = clScrollBar  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'TChart Component')  BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  RightAxis.Grid.Color = clScrollBar  ScaleLastPage = False  TopAxis.Grid.Color = clScrollBar  OnPageChange =  Chart1PageChange  Align = alClient  TabOrder = 0  <> object LineSeries1:	PixelsPerInch = 96
Left = 194 Top = 0 Width = 406 Height = 379 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	TextHeight = 13
Top = 0 Width = 406 Height = 379 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	<> object Chart1: TChart
Width = 406 Height = 379 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	Left = 194
Width = 406 Height = 379 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	Top = 0
BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = (     'TChart Component') BottomAxis.Grid.Color = clScrollBar LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	
BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = (     'TChart Component') BottomAxis.Grid.Color = clScrollBar LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	Height = 379
Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = (     'TChart Component') BottomAxis.Grid.Color = clScrollBar LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	BackWall.Brush.Color = clWhite
Title.Frame.Color = clScrollBar Title.Text.Strings = (     'TChart Component') BottomAxis.Grid.Color = clScrollBar LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	BackWall.Brush.Style = bsClear
Title.Text.Strings = (     'TChart Component')  BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  RightAxis.Grid.Color = clScrollBar  ScaleLastPage = False  TopAxis.Grid.Color = clScrollBar  OnPageChange = Chart1PageChange  Align = alClient  TabOrder = 0  <> object LineSeries1:	Foot.Frame.Color = clScrollBar
Title.Text.Strings = (     'TChart Component')  BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  RightAxis.Grid.Color = clScrollBar  ScaleLastPage = False  TopAxis.Grid.Color = clScrollBar  OnPageChange = Chart1PageChange  Align = alClient  TabOrder = 0  <> object LineSeries1:	
'TChart Component')  BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  RightAxis.Grid.Color = clScrollBar  ScaleLastPage = False  TopAxis.Grid.Color = clScrollBar  OnPageChange = Chart1PageChange  Align = alClient  TabOrder = 0  <> object LineSeries1:	
BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  RightAxis.Grid.Color = clScrollBar  ScaleLastPage = False  TopAxis.Grid.Color = clScrollBar  OnPageChange = Chart1PageChange  Align = alClient  TabOrder = 0  <> object LineSeries1:	
clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  RightAxis.Grid.Color = clScrollBar  ScaleLastPage = False  TopAxis.Grid.Color = clScrollBar  OnPageChange = Chart1PageChange  Align = alClient  TabOrder = 0  <> object LineSeries1:	
LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	
Legend.Alignment = laTop RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	
RightAxis.Grid.Color = clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	
clScrollBar ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	
ScaleLastPage = False TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	
TopAxis.Grid.Color = clScrollBar OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	
OnPageChange = Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	
Chart1PageChange Align = alClient TabOrder = 0 <> object LineSeries1:	
Align = alClient TabOrder = 0 <> object LineSeries1:	
TabOrder = 0 <> object LineSeries1:	
<> object LineSeries1:	
	TLineSeries

<u> </u>
ParentFont = False
TabOrder = 0
OnClick = BitBtn2Click
Glyph.Data = {
NumGlyphs = 2
end
<> object BitBtn4: TBitBtn
Left = 358
Top = 4
Width = 89
Height = 33
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 2
Kind = bkClose
end
end
<> object Panel2: TPanel
Left = 0
Top = 217
Width = 546
Height = 41
Align = alTop
Caption = 'This is a TImage
Delphi Component with a Metafile
Image'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 2
end
<> object SaveDialog1:
TSaveDialog
DefaultExt = 'WMF'
FileName = 'metademo.wmf'
Filter = 'Metafile Format *.WMF'
Options = [ofOverwritePrompt]
Left = 196
Top = 49
10p 7/

Ton = 160
Top = 160
end
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 194
Height = 379
Align = alLeft
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 1
<> object Label2: TLabel
Left = 11
Top = 10
Width = 104
Height = 13
HelpType = htKeyword
Caption = '&Num Points Per
Page:
FocusControl = SpinEdit1
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
end <> object Label1: TLabel
<> object Label1: TLabel
<> object Label1: TLabel Left = 11
<> object Label1: TLabel     Left = 11     Top = 76
<> object Label1: TLabel     Left = 11     Top = 76     Width = 62
<> object Label1: TLabel     Left = 11     Top = 76     Width = 62     Height = 22
<> object Label1: TLabel
<> object Label1: TLabel Left = 11 Top = 76 Width = 62 Height = 22 HelpType = htKeyword Caption = 'Label1'
<> object Label1: TLabel Left = 11 Top = 76 Width = 62 Height = 22 HelpType = htKeyword Caption = 'Label1' Font.Charset =
<> object Label1: TLabel Left = 11 Top = 76 Width = 62 Height = 22 HelpType = htKeyword Caption = 'Label1' Font.Charset = DEFAULT_CHARSET
<> object Label1: TLabel Left = 11 Top = 76 Width = 62 Height = 22 HelpType = htKeyword Caption = 'Label1' Font.Charset = DEFAULT_CHARSET Font.Color = clBlack
<> object Label1: TLabel Left = 11 Top = 76 Width = 62 Height = 22 HelpType = htKeyword Caption = 'Label1' Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -19
<> object Label1: TLabel Left = 11 Top = 76 Width = 62 Height = 22 HelpType = htKeyword Caption = 'Label1' Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -19 Font.Name = 'Arial'
<> object Label1: TLabel Left = 11 Top = 76 Width = 62 Height = 22 HelpType = htKeyword Caption = 'Label1' Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -19

Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clRed
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = True
XValues.Name = 'X'
XValues.Nultiplier =
1.000000000000000000000000000000000000
XValues.Order = loAscending
C
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.00000000000000000
YValues.Order = loNone
Left = 311
Top = 156
end
<> object BubbleSeries1:
TBubbleSeries
ColorEachPoint = False
Marks.ArrowLength = 8
Marks.Frame.Visible = False
Marks.Transparent = True
Marks.Visible = False
SeriesColor = clGreen
Pointer.HorizSize = 26
Pointer.InflateMargins = True
Pointer.Style = psCircle
Pointer.Style = pseriele Pointer.VertSize = 26
Pointer.Visible = True
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.00000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.0000000000000000000
YValues.Order = loNone
RadiusValues.DateTime = False
RadiusValues.Name = 'Radius'
RadiusValues.Multiplier =
1.000000000000000000
RadiusValues.Order = loNone
Left = 395
t .

Glyph.Data = {
Layout = blGlyphRight
NumGlyphs = 2
end
<> object BLastPage:
TBitBtn
Left = 102
Top = 106
<b>Width = 79</b>
Height = 32
Caption = '&Last'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 3
OnClick = BLastPageClick
Glyph.Data = {
Layout = blGlyphRight
NumGlyphs = 2
end
<> object BFirstPage:
TBitBtn
Left = 14
Top = 106
Width = 79
Height = 32
Caption = '&First'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = [] ParentFont = False
TabOrder = 4
OnClick = BFirstPageClick
Glyph.Data = {
NumGlyphs = 2
end
<> object CheckBox1:
TCheckBox
Left = 14
Top = 44
Width = 131

1
end
<> object SpinEdit1:
TSpinEdit
Left = 121
<b>Top</b> = 7
Width = 58
Height = 22
MaxValue = 100000
MinValue = 0
TabOrder = 0
Value = 0
OnChange = SpinEdit1Change
end
<> object ButtonPrevious:
TBitBtn
Left = 14
Top = 149
<b>Width = 79</b>
Height = 32
Caption = '&Previous'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 1
OnClick = ButtonPreviousClick
Glyph.Data = {
NumGlyphs = 2
end
<> object ButtonNext:
TBitBtn
Left = 102
Top = 149
Width = 79
Height = 32
Caption = '&Next'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 2
OnClick = ButtonNextClick
Onone Buttom textonich

Top = 108  Width = 575  Height = 404  Caption = 'TeeChart. TChartShape Component Example'  Color = clBtnFace  Font.Charset =  DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  OldCreateOrder = True  Position = poScreenCenter  OnCreate = FormCreate  PixelsPerInch = 96  TextHeight = 13  <> object Chart1: TChart  Left = 153  Top = 57  Width = 414  Height = 313  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  Foot.Frame.Color = clScrollBar  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'TChart Component')  BottomAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  Legend.Visible = False  RightAxis.Grid.Color = clScrollBar  TopAxis.Grid.Color = clScrollBar  Align = alClient  TabOrder = 0  <> object LineSeries1:  TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clRed  Pointer.InflateMargins = True  Pointer.Style = psRectangle  Pointer.Visible = False  XValues.DateTime = True  XValues.Name = 'X'	
Height = 404 Caption = 'TeeChart. TChartShape Component Example' Color = clBtnFace Font.Charset = DEFAULT CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style =    OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 57 Width = 414 Height = 313 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	Top = 108
Caption = 'TeeChart. TChartShape Component Example' Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 57 Width = 414 Height = 313 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	Width = 575
TChartShape Component Example' Color = clBtnFace Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 57 Width = 414 Height = 313 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = (     'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	Height = 404
Color = clBtnFace Font.Charset = DEFAULT CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 57 Width = 414 Height = 313 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = (     'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar TopAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	Caption = 'TeeChart.
Font.Charset = DEFAULT CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96  TextHeight = 13  <> object Chart1: TChart  Left = 153  Top = 57  Width = 414  Height = 313  BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar  Title.Frame.Color = clScrollBar  Title.Frat.Strings = (  'TChart Component')  BottomAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  Legend.Visible = False  RightAxis.Grid.Color = clScrollBar  Align = alClient  TabOrder = 0  <> object LineSeries1: TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clRed  Pointer.InflateMargins = True  Pointer.Style = psRectangle  Pointer.Visible = False  XValues.DateTime = True	TChartShape Component Example'
DEFAULT CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 57 Width = 414 Height = 313 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Frat.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	Color = clBtnFace
Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 57 Width = 414 Height = 313 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	Font.Charset =
Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 57 Width = 414 Height = 313 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	DEFAULT CHARSET
Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 57 Width = 414 Height = 313 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = (     'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	_
Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 57 Width = 414 Height = 313 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	
Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 57 Width = 414 Height = 313 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	
OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 57 Width = 414 Height = 313 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	
Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 57 Width = 414 Height = 313 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	
OnCreate = FormCreate  PixelsPerInch = 96  TextHeight = 13  <> object Chart1: TChart  Left = 153  Top = 57  Width = 414  Height = 313  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  Foot.Frame.Color = clScrollBar  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'TChart Component')  BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  Legend.Visible = False  RightAxis.Grid.Color = clScrollBar  Align = alClient  TabOrder = 0  <> object LineSeries1:  TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clRed  Pointer.InflateMargins = True  Pointer.Style = psRectangle  Pointer.Visible = False  XValues.DateTime = True	
PixelsPerInch = 96  TextHeight = 13  <> object Chart1: TChart  Left = 153  Top = 57  Width = 414  Height = 313  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  Foot.Frame.Color = clScrollBar  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'TChart Component')  BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  Legend.Visible = False  RightAxis.Grid.Color = clScrollBar  TopAxis.Grid.Color = clScrollBar  Align = alClient  TabOrder = 0  <> object LineSeries1:  TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clRed  Pointer.InflateMargins = True  Pointer.Style = psRectangle  Pointer.Visible = False  XValues.DateTime = True	
TextHeight = 13  <> object Chart1: TChart  Left = 153  Top = 57  Width = 414  Height = 313  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  Foot.Frame.Color = clScrollBar  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'TChart Component')  BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  Legend.Visible = False  RightAxis.Grid.Color = clScrollBar  TopAxis.Grid.Color = clScrollBar  Align = alClient  TabOrder = 0  <> object LineSeries1:  TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clRed  Pointer.InflateMargins = True  Pointer.Style = psRectangle  Pointer.Visible = False  XValues.DateTime = True	
<pre>&lt;&gt; object Chart1: TChart    Left = 153    Top = 57    Width = 414    Height = 313    BackWall.Brush.Color = clWhite    BackWall.Brush.Style = bsClear    Foot.Frame.Color = clScrollBar    Title.Frame.Color = clScrollBar    Title.Text.Strings = (         'TChart Component')    BottomAxis.Grid.Color = clScrollBar    LeftAxis.Grid.Color = clScrollBar    Legend.Alignment = laTop    Legend.Visible = False    RightAxis.Grid.Color = clScrollBar    TopAxis.Grid.Color = clScrollBar    Align = alClient    TabOrder = 0    &lt;&gt; object LineSeries1:    TLineSeries    Marks.ArrowLength = 8    Marks.ArrowLength = 8    Marks.Visible = False    SeriesColor = clRed    Pointer.InflateMargins = True    Pointer.Style = psRectangle    Pointer.Visible = False    XValues.DateTime = True</pre>	
Left = 153  Top = 57  Width = 414  Height = 313  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  Foot.Frame.Color = clScrollBar  Title.Frame.Color = clScrollBar  Title.Text.Strings = (  'TChart Component')  BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  Legend.Visible = False  RightAxis.Grid.Color = clScrollBar  TopAxis.Grid.Color = clScrollBar  Align = alClient  TabOrder = 0  <> object LineSeries1:  TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clRed  Pointer.InflateMargins = True  Pointer.Style = psRectangle  Pointer.Visible = False  XValues.DateTime = True	
Top = 57 Width = 414 Height = 313 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar TopAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	
Width = 414 Height = 313 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = ( 'TChart Component') BottomAxis.Grid.Color = clScrollBar LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar TopAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	
Height = 313  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  Foot.Frame.Color = clScrollBar  Title.Frame.Color = clScrollBar  Title.Text.Strings = (     'TChart Component')  BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  Legend.Visible = False  RightAxis.Grid.Color = clScrollBar  TopAxis.Grid.Color = clScrollBar  Align = alClient  TabOrder = 0     <> object LineSeries1:  TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clRed  Pointer.InflateMargins = True  Pointer.Style = psRectangle  Pointer.Visible = False  XValues.DateTime = True	
BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = (     'TChart Component') BottomAxis.Grid.Color = clScrollBar Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar TopAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0     <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	11 21 2
BackWall.Brush.Style = bsClear Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = (     'TChart Component') BottomAxis.Grid.Color = clScrollBar LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar TopAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	<u> </u>
Foot.Frame.Color = clScrollBar Title.Frame.Color = clScrollBar Title.Text.Strings = (     'TChart Component') BottomAxis.Grid.Color = clScrollBar LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar TopAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	BackWall.Brush.Color = clWhite
Title.Frame.Color = clScrollBar Title.Text.Strings = (     'TChart Component') BottomAxis.Grid.Color = clScrollBar LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar TopAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	BackWall.Brush.Style = bsClear
Title.Text.Strings = (     'TChart Component')  BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  Legend.Visible = False  RightAxis.Grid.Color = clScrollBar  TopAxis.Grid.Color = clScrollBar  Align = alClient  TabOrder = 0     <> object LineSeries1:  TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clRed  Pointer.InflateMargins = True  Pointer.Style = psRectangle  Pointer.Visible = False  XValues.DateTime = True	Foot.Frame.Color = clScrollBar
'TChart Component')  BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  Legend.Visible = False  RightAxis.Grid.Color = clScrollBar  TopAxis.Grid.Color = clScrollBar  Align = alClient  TabOrder = 0  <> object LineSeries1:  TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clRed  Pointer.InflateMargins = True  Pointer.Style = psRectangle  Pointer.Visible = False  XValues.DateTime = True	Title.Frame.Color = clScrollBar
BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  Legend.Visible = False  RightAxis.Grid.Color = clScrollBar  TopAxis.Grid.Color = clScrollBar  Align = alClient  TabOrder = 0  <> object LineSeries1:  TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clRed  Pointer.InflateMargins = True  Pointer.Style = psRectangle  Pointer.Visible = False  XValues.DateTime = True	Title.Text.Strings = (
BottomAxis.Grid.Color = clScrollBar  LeftAxis.Grid.Color = clScrollBar  Legend.Alignment = laTop  Legend.Visible = False  RightAxis.Grid.Color = clScrollBar  TopAxis.Grid.Color = clScrollBar  Align = alClient  TabOrder = 0  <> object LineSeries1:  TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clRed  Pointer.InflateMargins = True  Pointer.Style = psRectangle  Pointer.Visible = False  XValues.DateTime = True	'TChart Component')
LeftAxis.Grid.Color = clScrollBar Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar TopAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	
Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar TopAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	clScrollBar
Legend.Alignment = laTop Legend.Visible = False RightAxis.Grid.Color = clScrollBar TopAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	LeftAxis.Grid.Color = clScrollBar
Legend.Visible = False RightAxis.Grid.Color = clScrollBar TopAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	
RightAxis.Grid.Color = clScrollBar TopAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	
clScrollBar TopAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	
TopAxis.Grid.Color = clScrollBar Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	
Align = alClient TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	
TabOrder = 0 <> object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	
<pre>&lt;&gt; object LineSeries1: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True</pre>	<u> </u>
TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clRed  Pointer.InflateMargins = True  Pointer.Style = psRectangle  Pointer.Visible = False  XValues.DateTime = True	
Marks.ArrowLength = 8  Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	· ·
Marks.Visible = False SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	
SeriesColor = clRed Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	
Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	
Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True	
Pointer.Visible = False XValues.DateTime = True	
XValues.DateTime = True	
XValues.Name = 'X'	
	XValues.Name = 'X'

77.1.1.40
Height = 19
Caption = 'Auto Scale Last Page'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 5
OnClick = CheckBox1Click
end
<> object BitBtn3: TBitBtn
Left = 40
Top = 332
Width = 93
Height = 33
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 6
Kind = bkClose
end
<> object Memo1: TMemo
Left = 32
Top = 200
Width = 121
Height = 109
Lines.Strings = (
'Using the Chart'
'MaxPointsPerPage'
'property Series can'
'be divided in "pages"'
'so you can navigate'
'across them.')
TabOrder = 7
end
end
End
111111111111111111111111111111111111111
<> object ShapesForm:
TShapesForm
Left = 200

Marks.Visible = False SeriesColor = clYellow Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	
Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	Marks.Visible = False
Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = True XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	SeriesColor = clYellow
Pointer.Visible = False XValues.DateTime = True XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	Pointer.InflateMargins = True
Pointer.Visible = False XValues.DateTime = True XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	Pointer.Style = psRectangle
XValues.DateTime = True XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	
XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	
XValues.Multiplier = 1.000000000000000000000000000000000000	
1.000000000000000000000000000000000000	
XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000000000000000000000	
YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000000000000000000000	
YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000000000000000000000	
YValues.Multiplier = 1.000000000000000000000000000000000000	
1.000000000000000000000000000000000000	
YValues.Order = loNone  Left = 345  Top = 130 end  <> object ChartShape2: TChartShape  Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clWhite Brush.Color = clRed Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'Arial' Font.Style = [] Pen.Color = clLime Style = chasLine X0 = 729090.000000000000000000000000000000000	_
Left = 345 Top = 130 end <> object ChartShape2: TChartShape Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clWhite Brush.Color = clRed Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'Arial' Font.Style = [] Pen.Color = clLime Style = chasLine X0 = 729090.000000000000000000000000000000000	
end  <> object ChartShape2: TChartShape  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clWhite  Brush.Color = clRed  Font.Charset =  DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'Arial'  Font.Style = []  Pen.Color = clLime  Style = chasLine  X0 = 729090.000000000000000000000000000000000	
end <> object ChartShape2: TChartShape Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clWhite Brush.Color = clRed Font.Charset = DEFAULT CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'Arial' Font.Style = [] Pen.Color = clLime Style = chasLine X0 = 729090.000000000000000000000000000000000	
<pre>C&gt; object ChartShape2: TChartShape    Marks.ArrowLength = 8    Marks.Visible = False    SeriesColor = clWhite    Brush.Color = clRed    Font.Charset =    DEFAULT_CHARSET    Font.Name = 'Arial'    Font.Style = []    Pen.Color = clLime    Style = chasLine    X0 =    729090.000000000000000000000000000000000</pre>	-
TChartShape  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clWhite  Brush.Color = clRed  Font.Charset =  DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'Arial'  Font.Style = []  Pen.Color = clLime  Style = chasLine  X0 =  729090.000000000000000000000000000000000	
Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clWhite  Brush.Color = clRed  Font.Charset =  DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'Arial'  Font.Style = []  Pen.Color = clLime  Style = chasLine  X0 =  729090.000000000000000000000000000000000	¥
Marks.Visible = False SeriesColor = clWhite Brush.Color = clRed Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'Arial' Font.Style = [] Pen.Color = clLime Style = chasLine X0 = 729090.000000000000000000000000000000000	
SeriesColor = clWhite	Marks.ArrowLength = 8
Brush.Color = clRed Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'Arial' Font.Style = [] Pen.Color = clLime Style = chasLine X0 = 729090.000000000000000000000000000000000	Marks.Visible = False
Font.Charset = DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'Arial'  Font.Style = []  Pen.Color = clLime  Style = chasLine  X0 = 729090.000000000000000000000000000000000	SeriesColor = clWhite
DEFAULT_CHARSET	Brush.Color = clRed
Font.Color = clBlack Font.Height = -11 Font.Name = 'Arial' Font.Style = [] Pen.Color = clLime Style = chasLine X0 = 729090.000000000000000000000000000000000	Font.Charset =
Font.Name = 'Arial' Font.Style = [] Pen.Color = clLime Style = chasLine X0 = 729090.000000000000000000000000000000000	DEFAULT CHARSET
Font.Name = 'Arial' Font.Style = [] Pen.Color = clLime Style = chasLine X0 = 729090.000000000000000000000000000000000	Font.Color = clBlack
Font.Name = 'Arial' Font.Style = [] Pen.Color = clLime Style = chasLine X0 = 729090.000000000000000000000000000000000	Font.Height = -11
Font.Style = [] Pen.Color = clLime Style = chasLine X0 = 729090.000000000000000000000000000000000	
Pen.Color = clLime  Style = chasLine  X0 =  729090.000000000000000000000000000000000	
Style = chasLine	
X0 = 729090.000000000000000000000000000000000	
729090.000000000000000000000000000000000	
X1 = 729108.00000000000000000000000000000000000	
729108.00000000000000000000000000000000000	
Y0 = 692.75000000000000000000000000000000000000	
Y1 = 2915.500000000000000000000000000000000000	
XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000 XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000000000000000000000	
XValues.Name = 'X'  XValues.Multiplier = 1.000000000000000000000  XValues.Order = loAscending  YValues.DateTime = False  YValues.Name = 'Y'  YValues.Multiplier = 1.000000000000000000000000000000000000	
XValues.Multiplier = 1.000000000000000000000000000000000000	
1.000000000000000000000000000000000000	
XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000000000000000000000	
YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.00000000000000000 YValues.Order = loNone	
YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000000000000000000000	
YValues.Multiplier = 1.0000000000000000000 YValues.Order = loNone	
1.000000000000000000000000000000000000	
YValues.Order = loNone	
	1.0000000000000000000
Left = 161	YValues.Order = loNone
	Left = 161

XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 391
Top = 130
end
<> object ChartShape3:
TChartShape
Cursor = crCross
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clWhite
OnClick = ChartShape3Click
Brush.Color = clWhite
Brush.Style = bsDiagCross
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'Arial'
Font.Style = []
Style = chasRectangle
X0 =
729090.0000000000000000000
X1 =
729108.0000000000000000000
Y0 = 549.5000000000000000000
Y1 = 1742.0000000000000000000
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.0000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.0000000000000000000
YValues.Order = loNone
Left = 161
Top = 206
end
<> object LineSeries2:
TLineSeries
Marks.ArrowLength = 8
<del> </del>

Y 6, 40
Left = 19
Top = 4
Width = 402
Height = 19
HelpType = htKeyword
Caption = 'The Line and Ellipse
are TChartShape components.'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clNavy
Font.Height = -17
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
end
<> object Label2: TLabel
Left = 19
Top = 28
Width = 426
Height = 19
HelpType = htKeyword
Caption = 'Zoom or Scroll to see
how shapes follow Chart Series.'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clNavy
Font.Height = -17
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
end
end
<> object Panel2: TPanel
Left = 0
Top = 57
Width = 153
Height = 313
Align = alLeft
TabOrder = 2
<> object BitBtn2: TBitBtn
Left = 26
Top = 256
Width = 89
Height = 33
Caption = 'Close'
TabOrder = 0
Kind = bkClose
end

Top = 158
end
<> object ChartShape1:
TChartShape
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clWhite
Brush.Color = 8454016
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -21
Font.Name = 'Arial'
Font.Style = []
Text.Strings = (
'Hello'
'Tee'
'World !')
Pen.Color = clNavy
Pen.Width = 2
X0 =
729090.0000000000000000000
X1 =
729108.000000000000000000
Y0 = 737.25000000000000000000000000000000000000
Y1 = 3803.50000000000000000000000000000000000
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000000000000000000000
XValues.Order = loAscending
B
YValues.DateTime = False YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 157
Top = 108
end
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 567
Height = 57
Align = alTop
TabOrder = 1
<> object Label1: TLabel
UDIECT PADEII. I PADEI

# <u>AC.&P</u>

example of event handling.'
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Color = 8454016
BackWall.Pen.Visible = False
BottomWall.Color = clRed
Foot.Font.Charset =
DEFAULT CHARSET
Foot.Font.Color = clRed
Foot.Font.Height = -15
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsBold]
Foot.Frame.Color = clScrollBar
LeftWall.Color = 16777088
Title.Alignment = taLeftJustify
Title.Brush.Color = clWhite
Title.Brush.Style = bsClear
Title.Color = 16777088
Title.Font.Charset =
DEFAULT_CHARSET
Title.Font.Color = clBlue
Title.Font.Height = -16
Title.Font.Name = 'Arial'
Title.Font.Style = [fsBold, fsItalic]
Title.Frame.Color = 8421440
Title.Frame.Width = 4
Title.Text.Strings = (
'Pyramid Bar Chart built from
SQL Query.')
OnClickLegend =
DBChart1ClickLegend
BackColor = 8454016
BottomAxis.Grid.Color =
clScrollBar
Chart3DPercent = 40
ClipPoints = False
Frame.Visible = False
LeftAxis.Grid.Color = clScrollBar
Legend.Color = clAqua
Legend.Font.Charset =
DEFAULT_CHARSET
Legend.Font.Color = clBlack
Legend.Font.Height = -16
Legend.Font.Name = 'Arial'
Legend.Font.Style = [fsBold,
fsItalic]
Legend.Frame.Color = clTeal
Legend.Frame.Width = 4
Legend.ShadowSize = 11

<> object Memo1: TMemo
Left = 16
Top = 44
Width = 121
Height = 161
Lines.Strings = (
'Shapes are simple '
'Series components '
'with special formatting '
'attributes.'
11
'Zoom and scrolling'
'is also possible with'
'Shapes.')
TabOrder = 1
end
end
End
10
10
s abject COL Dang Forms
<> object SQLBarsForm:
TSQLBarsForm
Left = 191
Top = 116
Width = 584
Height = 471
Caption = 'TeeChart SQL Demo'
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
Scaled = False
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object DBChart1:
TDBChart
Left = 0
Top = 85
Width = 576
Height = 181
Hint = 'Click on a Bar to see an
mint - Chek un a dar to see an

Fort Color - alDod
Font.Color = clRed
Font.Height = -12
Font.Name = 'Arial'
Font.Style = [fsItalic]
ParentFont = False
TabOrder = 1
<> object BitBtn1: TBitBtn
Left = 292
Top = 4
Width = 89
Height = 33
TabOrder = 0
Kind = bkClose
end
end
<> object Panel2: TPanel
Left = 0
Top = 0
Width = 576
Height = 85
Align = alTop
TabOrder = 2
<> object Label1: TLabel
Left = 286
Top = 54
Width = 45
Height = 13
HelpType = htKeyword
Caption = 'Bar &Style:'
end
<> object Memo1: TMemo
Left = 16
Top = 10
Width = 261
Height = 67
Lines.Strings = (
'select Name,Weight from
":dbdemos:animals.dbf"'
'where Weight < 20'
")
TabOrder = 0
end
<> object BitBtn2: TBitBtn
Left = 286
Top = 10
Width = 111
Height = 33
Caption = 'Run Query !!'
Capuon - Kun Query ::

Legend.TextStyle = ltsRightValue
Legend.TopPos = 5
RightAxis.Grid.Color =
clScrollBar
TopAxis.Grid.Color = clScrollBar
Align = alClient
ParentShowHint = False
ShowHint = True
TabOrder = 0
<> object BarSeries1:
TBarSeries
ColorEachPoint = True
Cursor = crUpArrow
Marks.ArrowLength = 20
Marks.Style = smsPercent
Marks.Visible = True
DataSource = Query1
SeriesColor = clRed
XLabelsSource = 'NAME'
OnClick = BarSeries1Click
BarStyle = bsPyramid
BarWidthPercent = 80
XValues.DateTime = False
XValues.Name = 'X'
XValues.Nultiplier =
1.000000000000000000000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Nultiplier =
1.000000000000000000000000000000000000
YValues.Order = loNone
YValues.ValueSource =
'WEIGHT'
OnGetBarStyle =
BarSeries1GetBarStyle
Left = 82
Top = 118
end
end
<> object Panel1: TPanel
Left = 0
Top = 396
Width = 576
Height = 41
Align = alBottom
Alignment = taLeftJustify
Font.Charset =
DEFAULT_CHARSET

values !!!'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clNavy
Font.Height = -16
Font.Name = 'Arial'
Font.Style = [fsBold, fsItalic]
ParentFont = False
end
<> object DBGrid1:
TDBGrid
Left = 1
Top = 1
Width = 176
Height = 128
Align = alLeft
DataSource = DataSource1
TabOrder = 0
TitleFont.Charset =
DEFAULT CHARSET
TitleFont.Color = clBlack
TitleFont.Height = -11
TitleFont.Name = 'MS Sans
Serif'
TitleFont.Style = [] end
end
<> object DataSource1:
TDataSource
DataSet = Query1
Left = 215
Top = 36
end
<> object Query1: TQuery
Active = True
DatabaseName = 'DBDEMOS'
RequestLive = True
SQL.Strings = (
'select Name, Weight from '
"":dbdemos:animals.dbf"'
'where Weight<20'
")
Left = 118
Top = 34
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
11

TabOrder = 1 OnClick = BitBtn2Click Glyph.Data = { NumGlyphs = 2 end <> object ComboBox1:
Glyph.Data = {     NumGlyphs = 2     end <> object ComboBox1:
NumGlyphs = 2 end <> object ComboBox1:
end <> object ComboBox1:
end <> object ComboBox1:
•
•
TComboBox
Left = 342
Top = 50
Width = 127
Height = 21
Style = csDropDownList
ItemHeight = 13
TabOrder = 2
OnChange = ComboBox1Change
Items.Strings = (
'Rectangle'
'Pyramid'
'InvPyramid'
'Cilinder'
'Ellipse'
'Arrow')
end
<> object CBRandomBar:
TCheckBox
Left = 477
Top = 53
Width = 88
Height = 17
Caption = '< &Random'
TabOrder = 3
OnClick = CBRandomBarClick
end
end
<> object Panel3: TPanel
Left = 0
Top = 266
Width = 576
Height = 130
Align = alBottom
TabOrder = 3
<> object Label2: TLabel
Left = 184
Top = 22
Width = 203
Height = 19
HelpType = htKeyword
Caption = '< Try to change

# <u>AC.&P</u>

_
LeftAxis.Ticks.Color = clYellow
LeftAxis.Title.Caption =
'Quantity'
Legend.Font.Charset =
DEFAULT_CHARSET
Legend.Font.Color = clBlack
Legend.Font.Height = -12
Legend.Font.Name = 'Arial'
Legend.Font.Style = []
Align = alClient
BevelInner = bvRaised
BevelOuter = bvNone
BorderWidth = 14
TabOrder = 0
<> object BarSeries1:
TBarSeries
Marks.ArrowLength = 20
Marks.Visible = False
SeriesColor = clRed
MultiBar = mbNone
XValues.DateTime = True
XValues.Name = 'X'
XValues.Nultiplier =
1.000000000000000000000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier = 1.00000000000000000000000000000000000
YValues.Order = loNone
Left = 108
Top = 4
end
<> object BarSeries2:
TBarSeries
Marks.ArrowLength = 20
Marks.Visible = False
SeriesColor = clGreen
MultiBar = mbNone
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.00000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone

<> object StackedForm:
TStackedForm
Left = 191
Top = 109
Width = 652
Height = 431
Caption = 'Stacked Bars'
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Chart1: TChart
Left = 0
Top = 41
Width = 510
Height = 356
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Color = 8454016
BottomWall.Color = 16512
Foot.Font.Charset =
DEFAULT_CHARSET
Foot.Font.Color = clRed
Foot.Font.Height = -12
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsItalic]
Foot.Frame.Color = clScrollBar
Gradient.Direction = gdRightLeft
Gradient.StartColor = 16744576
LeftWall.Color = clBlue
Title.Frame.Color = clScrollBar
Title.Text.Strings = (
'TChart Component')
BackColor = 8454016
BottomAxis.DateTimeFormat =
'M/d/yy'
BottomAxis.Title.Caption = 'Sales
Date'
Chart3DPercent = 60
LeftAxis.MinorTickCount = 1
LeftAxis.MinorTickLength = 5
LeftAxis.MinorTicks.Color =
clBlue
LeftAxis.TickLength = 8

<> object Label1: TLabel
Left = 7
Top = 5
Width = 32
Height = 13
HelpType = htKeyword
Caption = '&Series:'
FocusControl = ComboBox1
end
<> object Shape1: TShape
Left = 7
Top = 52
Width = 118
Height = 29
HelpType = htKeyword
OnMouseUp = Shape1MouseUp
end
<> object Button1: TButton
Left = 17
Top = 86
Width = 99
Height = 29
Caption = 'S&croll Series'
TabOrder = 0
OnClick = Button1Click
end
<> object RadioGroup2:
TRadioGroup
Left = 13
Top = 150
Width = 108
Height = 139
Caption = '&Bar Style:'
ItemIndex = 0
Items.Strings = (
'Rectangle'
'Pyramid'
'InvPyramid'
'Cilinder'
'Ellipse'
'Arrow'
'Gradient')
TabOrder = 1
OnClick = RadioGroup2Click
end
<> object ComboBox1:
TComboBox
Left = 9

Left = 141 Top = 4 end  <> object BarSeries3: TBarSeries  Marks.ArrowLength = 20 Marks.Visible = False SeriesColor = clAqua MultiBar = mbNone XValues.DateTime = True XValues.Multiplier = 1.000000000000000000000000000000000000	
end  <> object BarSeries3: TBarSeries  Marks.ArrowLength = 20  Marks.Visible = False SeriesColor = clAqua  MultiBar = mbNone  XValues.DateTime = True  XValues.Name = 'X'  XValues.Multiplier = 1.000000000000000000000000000000000000	Left = 141
<pre> &lt;&gt; object BarSeries3: TBarSeries  Marks.ArrowLength = 20  Marks.Visible = False SeriesColor = clAqua MultiBar = mbNone XValues.DateTime = True XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000</pre>	Top = 4
TBarSeries  Marks.ArrowLength = 20  Marks.Visible = False  SeriesColor = clAqua  MultiBar = mbNone  XValues.DateTime = True  XValues.Name = 'X'  XValues.Multiplier = 1.000000000000000000000000000000000000	end
Marks.ArrowLength = 20 Marks.Visible = False SeriesColor = clAqua MultiBar = mbNone XValues.DateTime = True XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	<> object BarSeries3:
Marks.Visible = False SeriesColor = clAqua MultiBar = mbNone XValues.DateTime = True XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	TBarSeries
Marks.Visible = False SeriesColor = clAqua MultiBar = mbNone XValues.DateTime = True XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	Marks.ArrowLength = 20
MultiBar = mbNone XValues.DateTime = True XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	
MultiBar = mbNone XValues.DateTime = True XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	SeriesColor = clAqua
XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	
XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	XValues.DateTime = True
XValues.Multiplier = 1.000000000000000000000000000000000000	
1.000000000000000000000000000000000000	
XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.0000000000000000000 YValues.Order = loNone Left = 172 Top = 4 end end <> object RadioGroup1: TRadioGroup Left = 0 Top = 0 Width = 644 Height = 41 Align = alTop Caption = 'Bar &Position:' Columns = 4 ItemIndex = 0 Items.Strings = ( 'Behind' 'Side to Side' 'Stacked' 'Stacked 100%') TabOrder = 1 OnClick = RadioGroup1Click end <> object Panel1: TPanel Left = 510 Top = 41 Width = 134 Height = 356 Align = alRight	_
YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000000000000000000000	
YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000000000000000000000	ě
YValues.Multiplier = 1.000000000000000000000000000000000000	
1.000000000000000000  YValues.Order = loNone  Left = 172  Top = 4  end  end  <> object RadioGroup1:  TRadioGroup  Left = 0  Top = 0  Width = 644  Height = 41  Align = alTop  Caption = 'Bar &Position:'  Columns = 4  ItemIndex = 0  Items.Strings = (  'Behind'  'Side to Side'  'Stacked'  'Stacked'  'Stacked 100%')  TabOrder = 1  OnClick = RadioGroup1Click end  <> object Panel1: TPanel  Left = 510  Top = 41  Width = 134  Height = 356  Align = alRight	
YValues.Order = loNone Left = 172 Top = 4 end end <> object RadioGroup1: TRadioGroup Left = 0 Top = 0 Width = 644 Height = 41 Align = alTop Caption = 'Bar & Position:' Columns = 4 ItemIndex = 0 Items.Strings = ( 'Behind' 'Side to Side' 'Stacked' 'Stacked' 'Stacked 100%') TabOrder = 1 OnClick = RadioGroup1Click end <> object Panel1: TPanel Left = 510 Top = 41 Width = 134 Height = 356 Align = alRight	<u> </u>
Left = 172 Top = 4 end end <> object RadioGroup1: TRadioGroup Left = 0 Top = 0 Width = 644 Height = 41 Align = alTop Caption = 'Bar & Position:' Columns = 4 ItemIndex = 0 Items.Strings = ( 'Behind' 'Side to Side' 'Stacked' 'Stacked' 'Stacked 100%') TabOrder = 1 OnClick = RadioGroup1Click end <> object Panel1: TPanel Left = 510 Top = 41 Width = 134 Height = 356 Align = alRight	
rop = 4 end end <> object RadioGroup1: TRadioGroup Left = 0 Top = 0 Width = 644 Height = 41 Align = alTop Caption = 'Bar & Position:' Columns = 4 ItemIndex = 0 Items.Strings = ( 'Behind' 'Side to Side' 'Stacked' 'Stacked' 'Stacked 100%') TabOrder = 1 OnClick = RadioGroup1Click end <> object Panel1: TPanel Left = 510 Top = 41 Width = 134 Height = 356 Align = alRight	
end  <> object RadioGroup1: TRadioGroup  Left = 0  Top = 0  Width = 644  Height = 41  Align = alTop  Caption = 'Bar & Position:'  Columns = 4  ItemIndex = 0  Items.Strings = (     'Behind'     'Side to Side'     'Stacked'     'Stacked 100%')  TabOrder = 1  OnClick = RadioGroup1Click end  <> object Panel1: TPanel  Left = 510  Top = 41  Width = 134  Height = 356  Align = alRight	
end <> object RadioGroup1: TRadioGroup Left = 0 Top = 0 Width = 644 Height = 41 Align = alTop Caption = 'Bar &Position:' Columns = 4 ItemIndex = 0 Items.Strings = ( 'Behind' 'Side to Side' 'Stacked' 'Stacked 100%') TabOrder = 1 OnClick = RadioGroup1Click end <> object Panel1: TPanel Left = 510 Top = 41 Width = 134 Height = 356 Align = alRight	
<pre>&lt;&gt; object RadioGroup1: TRadioGroup Left = 0 Top = 0 Width = 644 Height = 41 Align = alTop Caption = 'Bar &amp; Position:' Columns = 4 ItemIndex = 0 Items.Strings = (     'Behind'     'Side to Side'     'Stacked'     'Stacked 100%') TabOrder = 1 OnClick = RadioGroup1Click end &lt;&gt; object Panel1: TPanel Left = 510 Top = 41 Width = 134 Height = 356 Align = alRight</pre>	
TRadioGroup  Left = 0  Top = 0  Width = 644  Height = 41  Align = alTop  Caption = 'Bar &Position:'  Columns = 4  ItemIndex = 0  Items.Strings = (     'Behind'     'Side to Side'     'Stacked'     'Stacked 100%')  TabOrder = 1  OnClick = RadioGroup1Click end  <> object Panel1: TPanel  Left = 510  Top = 41  Width = 134  Height = 356  Align = alRight	
Left = 0 Top = 0 Width = 644 Height = 41 Align = alTop Caption = 'Bar & Position:' Columns = 4 ItemIndex = 0 Items.Strings = ( 'Behind' 'Side to Side' 'Stacked' 'Stacked 100%') TabOrder = 1 OnClick = RadioGroup1Click end <> object Panel1: TPanel Left = 510 Top = 41 Width = 134 Height = 356 Align = alRight	
Top = 0  Width = 644  Height = 41  Align = alTop  Caption = 'Bar &Position:'  Columns = 4  ItemIndex = 0  Items.Strings = (    'Behind'    'Side to Side'    'Stacked'    'Stacked 100%')  TabOrder = 1  OnClick = RadioGroup1Click end  <> object Panel1: TPanel  Left = 510  Top = 41  Width = 134  Height = 356  Align = alRight	•
Width = 644 Height = 41 Align = alTop Caption = 'Bar &Position:' Columns = 4 ItemIndex = 0 Items.Strings = ( 'Behind' 'Side to Side' 'Stacked' 'Stacked 100%') TabOrder = 1 OnClick = RadioGroup1Click end <> object Panel1: TPanel Left = 510 Top = 41 Width = 134 Height = 356 Align = alRight	
Height = 41 Align = alTop Caption = 'Bar &Position:' Columns = 4 ItemIndex = 0 Items.Strings = (     'Behind'     'Side to Side'     'Stacked'     'Stacked 100%') TabOrder = 1 OnClick = RadioGroup1Click end <> object Panel1: TPanel Left = 510 Top = 41 Width = 134 Height = 356 Align = alRight	
Align = alTop Caption = 'Bar &Position:' Columns = 4 ItemIndex = 0 Items.Strings = (     'Behind'     'Side to Side'     'Stacked'     'Stacked 100%') TabOrder = 1 OnClick = RadioGroup1Click end <> object Panel1: TPanel Left = 510 Top = 41 Width = 134 Height = 356 Align = alRight	
Caption = 'Bar & Position:' Columns = 4 ItemIndex = 0 Items.Strings = (     'Behind'     'Side to Side'     'Stacked'     'Stacked 100%') TabOrder = 1 OnClick = RadioGroup1Click end <> object Panel1: TPanel Left = 510 Top = 41 Width = 134 Height = 356 Align = alRight	-
Columns = 4  ItemIndex = 0  Items.Strings = (     'Behind'     'Side to Side'     'Stacked'     'Stacked 100%')  TabOrder = 1  OnClick = RadioGroup1Click end     <> object Panel1: TPanel  Left = 510  Top = 41  Width = 134  Height = 356  Align = alRight	
ItemIndex = 0 Items.Strings = ( 'Behind' 'Side to Side' 'Stacked' 'Stacked 100%') TabOrder = 1 OnClick = RadioGroup1Click end <> object Panel1: TPanel Left = 510 Top = 41 Width = 134 Height = 356 Align = alRight	
Items.Strings = (     'Behind'     'Side to Side'     'Stacked'     'Stacked 100%')     TabOrder = 1     OnClick = RadioGroup1Click     end     <> object Panel1: TPanel     Left = 510     Top = 41     Width = 134     Height = 356     Align = alRight	
'Behind' 'Side to Side' 'Stacked' 'Stacked 100%')  TabOrder = 1 OnClick = RadioGroup1Click end <> object Panel1: TPanel Left = 510 Top = 41 Width = 134 Height = 356 Align = alRight	
'Side to Side' 'Stacked' 'Stacked 100%')  TabOrder = 1  OnClick = RadioGroup1Click end <> object Panel1: TPanel Left = 510  Top = 41  Width = 134  Height = 356  Align = alRight	
'Stacked' 'Stacked 100%')  TabOrder = 1 OnClick = RadioGroup1Click end <> object Panel1: TPanel Left = 510 Top = 41 Width = 134 Height = 356 Align = alRight	
'Stacked 100%')  TabOrder = 1  OnClick = RadioGroup1Click end <> object Panel1: TPanel  Left = 510  Top = 41  Width = 134  Height = 356  Align = alRight	
TabOrder = 1 OnClick = RadioGroup1Click end <> object Panel1: TPanel Left = 510 Top = 41 Width = 134 Height = 356 Align = alRight	
OnClick = RadioGroup1Click end <> object Panel1: TPanel Left = 510 Top = 41 Width = 134 Height = 356 Align = alRight	,
end <> object Panel1: TPanel  Left = 510  Top = 41  Width = 134  Height = 356  Align = alRight	
<> object Panel1: TPanel  Left = 510  Top = 41  Width = 134  Height = 356  Align = alRight	
Left = 510 Top = 41 Width = 134 Height = 356 Align = alRight	
Top = 41 Width = 134 Height = 356 Align = alRight	•
Width = 134 Height = 356 Align = alRight	
Height = 356 Align = alRight	Top = 41
Align = alRight	
	Height = 356
TabOrder = 2	Align = alRight
	TabOrder = 2

Interval = 1
OnTimer = Timer1Timer
Left = 452
Top = 18
end
End
\\\\\\\
12
<> object TablePieForm:
TTablePieForm
Left = 193
Top = 109
Width = 629
Height = 440
Caption = 'TeeChart Table-Pie
Demo'
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -9
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
PixelsPerInch = 96
TextHeight = 13
<> object DBChart1:
TDBChart
Left = 0
Top = 0
Width = 621
Height = 255
AllowZnown = Folia
AllowZoom = False
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Pen.Visible = False
Foot.Font.Charset =
DEFAULT_CHARSET
Foot.Font.Color = clRed
Foot.Font.Height = -15
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsBold]
Foot.Frame.Color = clScrollBar
Foot.Text.Strings = (

Ton - 22
Top = 23 Width = 116
Height = 21
Style = csDropDownList
ItemHeight = 13
TabOrder = 2
OnChange = ComboBox1Change
end
<> object CheckBox1:
TCheckBox
Left = 9
Top = 123
Width = 120
Height = 19
Caption = '&Animate !!!'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clRed
Font.Height = -19
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 3
OnClick = CheckBox1Click
end
<> object BitBtn3: TBitBtn
Left = 25
Top = 320
Width = 89
Height = 33
TabOrder = 4
Kind = bkClose
end
<> object CheckBox2:
TCheckBox
Left = 43
Top = 296
Width = 50
Height = 17
Caption = '3D'
Checked = True
State = cbChecked
TabOrder = 5
OnClick = CheckBox2Click
end
end
<> object Timer1: TTimer
Enabled = False
Eliableu – Paise

No. 1 D. 1 G. 1 WYN 1
Marks.BackColor = clWhite
Marks.Font.Charset =
DEFAULT_CHARSET
Marks.Font.Color = clRed
Marks.Font.Height = -13
Marks.Font.Name = 'Arial'
Marks.Font.Style = [fsItalic]
Marks.Frame.Color = clNavy
Marks.Frame.Width = 3
Marks.Style = smsLabelValue
Marks.Visible = True
DataSource = Table1
SeriesColor = clRed
XLabelsSource = 'NAME'
OnClick = PieSeries1Click
Circled = True
OtherSlice.Text = 'Other'
PieValues.DateTime = False
PieValues.Name = 'Y'
PieValues.Multiplier =
1.000000000000000000
PieValues.Order = loNone
PieValues.ValueSource = 'SIZE'
Left = 187
Top = 6
end
end
<> object Panel2: TPanel
Left = 0
Top = 255
Width = 621
Height = 117
Align = alBottom TabOrder = 1
<> object DBGrid1:
TDBGrid
Left = 349
Top = 1
Width = 271
Height = 115
Align = alRight
DataSource = DataSource1
TabOrder = 0
TitleFont.Charset =
DEFAULT_CHARSET
TitleFont.Color = clBlack
TitleFont.Height = -9
TitleFont.Name = 'MS Sans

'XLabelSource:=Table1Name'
'PieValueSource:=Table1Size')
Title.Font.Charset =
DEFAULT_CHARSET
Title.Font.Color = clBlue
Title.Font.Height = -16
Title.Font.Name = 'Arial'
Title.Font.Style = [fsBold, fsItalic]
Title.Frame.Color = clScrollBar
Title.Text.Strings = (
'Pie Chart built from Table
:DBDEMOS:ANIMALS.DBF')
AxisVisible = False
BottomAxis.Grid.Color =
clScrollBar
ClipPoints = False
Frame.Visible = False
LeftAxis.Grid.Color = clScrollBar
Legend.Alignment = laLeft
Legend.Color = clAqua
Legend.Font.Charset =
DEFAULT_CHARSET
Legend.Font.Color = clBlack
Legend.Font.Height = -16
Legend.Font.Name = 'Arial'
Legend.Font.Style = [fsBold,
fsItalic]
Legend.Frame.Color = clTeal
Legend.Frame.Width = 4
Legend.ShadowColor = clGray
Legend.ShadowSize = 11
Legend.TextStyle = ltsLeftPercent
RightAxis.Grid.Color =
clScrollBar
TopAxis.Grid.Color = clScrollBar
View3DOptions.Elevation = 315
View3DOptions.Orthogonal =
False
View3DOptions.Perspective = 0
View3DOptions.Rotation = 360
View3DWalls = False
Align = alClient
TabOrder = 0
<> object PieSeries1:
TPieSeries
Cursor = crDrag
Marks.Arrow.Color = clYellow
Marks.Arrow.Width = 2
Marks.ArrowLength = 11

Left = 0
Top = 372
Width = 621
Height = 34
Align = alBottom
Alignment = taLeftJustify
Font.Charset =
DEFAULT CHARSET
Font.Color = clRed
Font.Height = -9
Font.Name = 'Arial'
Font.Style = [fsItalic]
ParentFont = False
TabOrder = 2
end
<> object Table1: TTable
Active = True
DatabaseName = 'DBDEMOS'
TableName = 'ANIMALS.DBF'
Left = 29
Top = 344
<> object Table1NAME:
TStringField
FieldName = 'NAME'
Size = 10
end
<> object Table1SIZE:
TSmallintField
FieldName = 'SIZE'
end
<> object Table1WEIGHT:
<b>TSmallintField</b>
FieldName = 'WEIGHT'
end
end
<> object DataSource1:
<b>TDataSource</b>
DataSet = Table1
Left = 111
Top = 344
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
13
<> object TeeMainForm:
TTeeMainForm
Left = 192
Top = 109

Serif'
TitleFont.Style = []
end
<> object CheckBox1:
TCheckBox
Left = 10
Top = 8
Width = 101
Height = 14
Caption = 'ACTIVE Table'
Checked = True
State = cbChecked
TabOrder = 1
OnClick = CheckBox1Click
end
<> object RadioGroup1:
TRadioGroup
Left = 10
Top = 32
Width = 131
Height = 32
Caption = 'Use Field:'
Columns = 2
ItemIndex = 0
Items.Strings = (
'Size'
'Weight')
TabOrder = 2
OnClick = RadioGroup1Click
end
<> object BitBtn1: TBitBtn
Left = 27
Top = 79
Width = 90
Height = 27
Caption = 'Close'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'Arial'
Font.Style = []
ParentFont = False
TabOrder = 3
Kind = bkClose
end
end
<> object Panel1: TPanel

<> object Panel1: TPanel
Left = 0
Top = 376
Width = 632
Height = 30
Align = alBottom
Alignment = taLeftJustify
TabOrder = 0
<> object Label11: TLabel
Left = 12
Top = 49
Width = 3
Height = 13
HelpType = htKeyword
end
<> object Label13: TLabel
Left = 12
Top = 6
Width = 129
Height = 19
HelpType = htKeyword
Caption = 'Steema Software'
Font.Charset =
DEFAULT CHARSET
Font.Color = clNavy
Font.Height = -16
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
end
<> object Label15: TLabel
3
Left = 212
Top = 9
Width = 27
Height = 13
HelpType = htKeyword
Caption = 'email:'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
<> object Label16: TLabel Left = 244
Top = 9

ActiveControl = Button2
BorderIcons = [biSystemMenu]
BorderStyle = bsDialog
Caption = 'TeeChart Standard
version 4.0 - Demo'
ClientHeight = 406
ClientWidth = 632
Color = clNavy
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Image1: TImage
Left = 484
Top = 26
Width = 141
Height = 45
HelpType = htKeyword
AutoSize = True
Picture.Data = {
OnClick = Label10Click
end
<> object Label2: TLabel
Left = 168
Top = 353
Width = 300
Height = 13
HelpType = htKeyword
Caption = '© Copyright 1996-2001
by David Berneda. All Rights
Reserved.'
Color = clNavy
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWhite
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentColor = False
ParentFont = False
end
<u> </u>

TabOrder = 1
<> object Button2: TButton
Left = 8
Top = -1
Width = 141
Height = 41
Caption = 'Show me the Demo!'
Default = True
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -12
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 0
OnClick = Button2Click
end
<> object Button3: TButton
Left = 8
Top = 85
Width = 141
Height = 29
Caption = 'Features'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -12
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 1
OnClick = Button3Click
end
<> object Button4: TButton
Left = 8
Top = 122
10p = 122 Width = 141
Height = 29
Caption = 'Pro Specifications'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -12
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 2

<b>Width</b> = <b>85</b>
Height = 13
HelpType = htKeyword
Caption = 'info@steema.com'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
<> object Label3: TLabel
Left = 375
Top = 6
Width = 119
Height = 16
Cursor = crHandPoint
HelpType = htKeyword
Caption = 'www.steema.com'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold, fsUnderline]
ParentFont = False
OnClick = Label3Click
end
<> object Button1: TButton
Left = 536
Top = 4
Width = 89
Height = 23
Caption = 'Close'
ModalResult = 1
TabOrder = 0
OnClick = Button1Click
end
end
<> object TPanel
Left = 476
Top = 102
Width = 157
Height = 251
BevelOuter = bvNone
BorderWidth = 1
Color = clNavy

Font.Charset =
DEFAULT_CHARSET
Font.Color = clRed
Font.Height = -29
Font.Name = 'Times New
Roman'
Font.Style = [fsBold]
ParentFont = False
Transparent = True
OnClick = Label10Click
OnMouseDown =
Chart2MouseDown
end
<> object FastLineSeries1:
TFastLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clAqua
LinePen.Color = clAqua
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
end
<> object FastLineSeries2:
<b>TFastLineSeries</b>
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clYellow
LinePen.Color = clYellow
XValues.DateTime = True
XValues.Date Time = True  XValues.Name = 'X'
XValues.Nultiplier =
1.000000000000000000000000000000000000
XValues.Order = loAscending
YValues Name = IV!
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000000000000000000000
YValues.Order = loNone
end
end
end

OnClick = Button4Click
end
<> object Button6: TButton
Left = 8
Top = 48
Width = 141
Height = 29
Caption = 'Chart Styles'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -12
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 3
OnClick = Button6Click
end
<> object Chart2: TChart
Left = 8
Top = 168
Width = 141
Height = 73
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Pen.Visible = False
Title.Text.Strings = (
'TChart')
Title.Visible = False
AxisVisible = False
ClipPoints = False
Frame.Visible = False
Legend.Color = clSilver
Legend.Visible = False
View3DWalls = False
BevelWidth = 2
Color = clSilver
TabOrder = 4
OnMouseDown =
Chart2MouseDown
OnMouseUp = Chart2MouseUp
<pre>&lt;&gt; object Label12: TLabel</pre>
·
Left = 25
Left = 25 Top = 16
Left = 25 Top = 16 Width = 97
Left = 25 Top = 16 Width = 97 Height = 32
Left = 25 Top = 16 Width = 97

# <u>AC.&P</u>

Marks.Visible = False
SeriesColor = clRed
CustomYRadius = 25
ExplodeBiggest = 25
OtherSlice.Text = 'Other'
PieValues.DateTime = False
PieValues.Name = 'Pie'
PieValues.Multiplier =
1.000000000000000000000000000000000000
PieValues.Order = loNone
end
end
<> object Chart8: TChart
Left = 3
Top = 163
<b>Width = 97</b>
Height = 66
AllowPanning = pmNone
AllowZoom = False
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Pen.Visible = False
MarginBottom = 10
MarginLeft = 5
MarginRight = 5
MarginTop = 0
Title.Font.Charset =
DEFAULT_CHARSET
Title.Font.Color = clBlack
Title.Font.Height = -17
Title.Font.Name = 'Arial'
Title.Font.Style = [fsBold]
Title.Text.Strings = (
")
AxisVisible = False
Chart3DPercent = 95
ClipPoints = False
Frame.Visible = False
Legend.Visible = False View 3 DOntions Elevation = 315
View3DOptions.Elevation = 315
View3DOptions.Orthogonal =
False
View3DOptions.Perspective = 0
View3DOptions.Rotation = 360
View3DOptions.Zoom = 87
View3DWalls = False
BevelOuter = bvNone
Color = clAqua
-

11 (D 10 ED 1
<> object Panel2: TPanel
Left = 9
Top = 102
Width = 465
Height = 241
BevelWidth = 2
Color = clAqua
TabOrder = 2
<> object Chart7: TChart
Left = 3
Top = 7
Width = 97
Height = 66
AllowPanning = pmNone
AllowZoom = False
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Pen.Visible = False
MarginBottom = 10
MarginLeft = 5
MarginRight = 5
MarginTop = 0
Title.Font.Charset =
DEFAULT_CHARSET
Title.Font.Color = clBlack
Title.Font.Height = -17
Title.Font.Name = 'Arial'
Title.Font.Style = [fsBold]
Title.Text.Strings = (
")
AxisVisible = False
Chart3DPercent = 50
ClipPoints = False
Frame.Visible = False
Legend.Visible = False
View3DOptions.Elevation = 315
View3DOptions.Orthogonal =
False
View3DOptions.Perspective = 0
View3DOptions.Rotation = 360
View3DOptions.Rotation = 300 View3DOptions.Zoom = 92
1
View3DWalls = False
BevelOuter = bvNone
Color = clAqua
TabOrder = 1
<> object PieSeries6:
TPieSeries
Marks.ArrowLength = 8

View3DOptions.Zoom = 92
View3DWalls = False
BevelOuter = bvNone
Color = clAqua
TabOrder = 4
<> object PieSeries9:
TPieSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clRed
CustomYRadius = 25
ExplodeBiggest = 40
OtherSlice.Text = 'Other'
PieValues.DateTime = False
PieValues.Name = 'Pie'
PieValues.Multiplier =
1.000000000000000000000000000000000000
PieValues.Order = loNone
end
end
<> object Chart9: TChart
Left = 206
Top = 4
Width = 255
Height = 221
AllowPanning = pmNone
AllowZoom = False
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Pen.Visible = False
MarginBottom = 10
MarginLeft = 5
MarginRight = 5
MarginTop = 0
Title.Font.Charset =
DEFAULT CHARSET
Title.Font.Color = clBlack
Title.Font.Height = -24
Title.Font.Name = 'Arial'
Title.Font.Style = [fsBold]
Title.Text.Strings = (
")
Title.Visible = False
AxisVisible = False
Clin Points - Folce
ClipPoints = False
Frame.Visible = False
Legend.TextStyle = ltsPlain

TabOrder = 2
<> object PieSeries7:
TPieSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clRed
CustomYRadius = 25
ExplodeBiggest = 20
OtherSlice.Text = 'Other'
PieValues.DateTime = False
PieValues.Name = 'Pie'
PieValues.Multiplier =
1.0000000000000000000
PieValues.Order = loNone
end
end
<> object Chart10: TChart
Left = 105
Top = 139
Width = 97
Height = 66
AllowPanning = pmNone AllowZoom = False
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Pen.Visible = False
MarginBottom = 10
MarginLeft = 5
MarginRight = 5
MarginTop = 0
Title.Font.Charset =
DEFAULT_CHARSET
Title.Font.Color = clBlack
Title.Font.Height = -17
Title.Font.Name = 'Arial'
Title.Font.Style = [fsBold]
Title.Text.Strings = (
")
AxisVisible = False
Chart3DPercent = 100
ClipPoints = False
Frame.Visible = False
Legend.Visible = False
View3DOptions.Elevation = 315
View3DOptions.Orthogonal =
False
View3DOptions.Perspective = 0
View3DOptions.Rotation = 360

Top = 31
Width = 97
Height = 66
AllowPanning = pmNone
AllowZoom = False
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Pen.Visible = False
MarginBottom = 10
MarginLeft = 5
MarginRight = 5
MarginTop = 0
Title.Font.Charset =
DEFAULT CHARSET
Title.Font.Color = clBlack
Title.Font.Height = -17
Title.Font.Name = 'Arial'
Title.Font.Style = [fsBold]
Title.Text.Strings = (
")
AxisVisible = False
Chart3DPercent = 50
ClipPoints = False
Frame.Visible = False
Legend.Visible = False
View3DOptions.Elevation = 315
View3DOptions.Orthogonal =
False
View3DOptions.Perspective = 0
View3DOptions.Rotation = 360
View3DWalls = False
BevelOuter = bvNone
Color = clAqua
TabOrder = 0
<> object Series1:
<b>TPieSeries</b>
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clRed
CustomYRadius = 25
ExplodeBiggest = 25 OtherSlice.Text = 'Other'
PieValues DateTime = False
PieValues.Name = 'Pie'
PieValues.Multiplier =
1.000000000000000000
PieValues.Order = loNone
end

Legend.Visible = False
View3DOptions.Elevation = 315
View3DOptions.Orthogonal =
False
View3DOptions.Perspective = 0
View3DOptions.Rotation = 360
View3DWalls = False
BevelOuter = bvNone
Color = clAqua
TabOrder = 3
<> object PieSeries8:
TPieSeries C. L. C
Marks.Arrow.Color = clBlack
Marks.ArrowLength = 1
Marks.Font.Charset =
DEFAULT_CHARSET
Marks.Font.Color = clNavy
Marks.Font.Height = -11
Marks.Font.Name = 'Arial'
Marks.Font.Style = []
Marks.Frame.Visible = False
Marks.Transparent = True
Marks.Visible = True
DataSource = Series1
PercentFormat = '##0.00 %'
SeriesColor = clRed
CustomYRadius = 35
ExplodeBiggest = 40
OtherSlice.Text = 'Other'
PiePen.Visible = False
PieValues.DateTime = False
PieValues.Name = 'Pie'
PieValues.Multiplier =
1.00000000000000000
PieValues.Order = loNone
DataSources = (
'Series1'
'PieSeries6'
'PieSeries7'
'PieSeries9')
,
<> object PieSeries8TAddTeeFunction:
TAddTeeFunction
end
end
end
<> object Chart1: TChart
Left = 105

Caption = 'For Borland Delphi,
Kylix and C++ Builder'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clNavy
Font.Height = -16
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
end
end
<> object Timer1: TTimer
Interval = 1
OnTimer = Timer1Timer
Left = 408
Top = 283
end
<> object Timer2: TTimer
Interval = 200
OnTimer = Timer2Timer
Left = 356
Top = 281
end
End
End
14
<> object AxisLabelsForm: TAxisLabelsForm
Left = 197
Top = 109
Width = 501
Height = 355
Caption = 'TeeChart Sample
Project. Custom Axis Labeling'
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Chart1: TChart
Left = 153
Top = 0
Width = 340
Height = 321
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
Dack it ani. Di usni. Style – Dscical

1
end
end
<> object TPanel
Left = 8
Top = 359
<b>Width = 156</b>
Height = 1
TabOrder = 3
end
<> object TPanel
Left = 475
Top = 360
Width = 153
Height = 1
TabOrder = 4
end
<> object Panel3: TPanel
Left = 8
Top = 11
Width = 465
Height = 81
TabOrder = 5
<> object Label1: TLabel
Left = 8
Top = 15
Width = 449
Height = 22
HelpType = htKeyword
Alignment = taCenter
AutoSize = False
Caption = 'The 100% Native
Charting VCL / CLX Library'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlue
Font.Height = -19
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
end
<> object Label9: TLabel
Left = 8
Top = 44
Width = 449
Height = 19
HelpType = htKeyword
Alignment = taCenter
AutoSize = False
THUODIZE I AISE

<> object PointSeries1:
TPointSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clGreen
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = True
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.0000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 283
Top = 132
end
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 153
Height = 321
Align = alLeft
TabOrder = 1
<> object RadioGroup1:
TRadioGroup
Left = 17
Top = 16
Width = 120
Height = 57
Caption = 'Choose Axis Labels:'
ItemIndex = 1
Items.Strings = (
'Default'
'User defined')
TabOrder = 0
OnClick = RadioGroup1Click
end
<> object BitBtn3: TBitBtn
Left = 32
Top = 328
10p - 328 Width = 89
Height = 33

Foot.Frame.Color = clScrollBar
Title.Frame.Color = clScrollBar
Title.Text.Strings = (
'TChart Component')
BottomAxis.LabelsFont.Charset =
DEFAULT_CHARSET
BottomAxis.LabelsFont.Color =
clMaroon
BottomAxis.LabelsFont.Height = -
21
BottomAxis.LabelsFont.Name =
'Arial'
BottomAxis.LabelsFont.Style =
[fsBold, fsItalic]
LeftAxis.LabelsFont.Charset =
DEFAULT_CHARSET  LeftAxis.LabelsFont.Color =
clNavy
LeftAxis.LabelsFont.Height = -24
LeftAxis.LabelsFont.Name =
'Arial'
LeftAxis.LabelsFont.Style =
[fsBold, fsItalic]
Legend.Alignment = laTop
OnGetNextAxisLabel =
Chart1GetNextAxisLabel
Align = alClient
TabOrder = 0
<> object LineSeries1:
TLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clRed
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = True
XValues.Name = 'X'
XValues.Nultiplier =
1.0000000000000000000000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 199
Top = 122
end

<> object Memo1: TMemo
Left = 0
Top = 167
Width = 467
Height = 307
Align = alClient
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
Lines.Strings = (
'Some of the TeeChart Pro 5.0
features are:'
"
'** Supports Kylix.'
'** Optional FULL 100%
source code and Pro-Support at
additional '+
'price'
'** Electronic delivery. Optional
CDRom and printed guide mailed.'
'** Multiple Charts per Form
and Multiple Series per Chart.'
'** New! Multiple unlimited
axes per Chart.'
*** New! 3D Rotation, elevation,
zoom and scroll.'
'** New! Optional OpenGL 3D
realistic rendering'
'** New ! Multi-row Legends'
'** New! Logarithmic axis
labels'
'** New! Radar, Surface 3D,
Contour 3D, Point 3D, and Error-
Seri' +
'es'
'** New! Pie Slice Exploding'
'** New! Custom Arrays
supported.'
'** Up to 250000000 points per
Series!'
'** Add, change, delete values
manually or'
'** Link directly to Tables and
Queries. (TDBChart)'

TabOrder = 1
Kind = bkClose
end
<> object Memo1: TMemo
Left = 16
Top = 96
Width = 121
Height = 193
Lines.Strings = (
'This example shows '
'how Axis labels can be '
'customized, '
'both their position and '
'text using '
'OnGetAxisLabel and'
'OnGetNextAxisLabel '
'events.'
11
'In this example Bottom '
'axis labels are set to the '
'starting day of each'
'month.')
TabOrder = 2
end
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
15
<> object ChartSpecs:
TChartSpecs
Left = 192
Top = 106 Width = 475
Height = 508
ActiveControl = Memo1
Caption = 'TeeChart Pro 5.0
Specifications'
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
PixelsPerInch = 96
TextHeight = 13
1 CALIFORNI - 15

Stacked B' +	
'ars'	'** Any Series can be linked to
'** Stacked 100%, Stack Groups'	another Series (both at Design an' +
'** Error Bars, Arrows'	'd RunTime).'
'** Bubble, Pie, Gantt, Candle-	'** Non Data-Aware charting if
Stick, Curve Fitting'	desired.
'** Waterfall, ColorGrid,	*** Seamless integration with
Histogram, Smith'	QuickReport 2.0 and 3.0'
'** Funnel, Bollinger, ADX,	'** Automatic Resize and
MACD, Pyramid, Donut, Box-	Refreshing'
Whisker'	'** Animated Zoom, Scrolling &
'** Automatic paging of Series	RealTime capabilities.'
points'	'** 2D 3D. Z-Order
'** Custom drawing and	configurable.'
printing'	'** Standard, Statistical &
'** Gradient and Picture	Financial Series types included.'
Background filling'	'** Extended Series types and
'** Many new samples and	Functions are easy to create.'
example projects'	'** Tones of properties to
'** PacMan game made with	customize your preferences.'
TeeChart.'	
'** No Royalties or Runtime	'** Events to control user''s
fees.'	input and override default
"	behaviour' +
140 0 115	's'
'(*) QuickReport is copyright	*** Resource messages (*.rc) to
by Allan Lochert, QuSoft AS,	allow internationalization'
Norway' +	'** Property Editor dialogs also
'.') ParentFont = False	available at RunTime.'
ReadOnly = True	'** Complete Printing control both to Printer and User Canvas.'
· ·	'** Remote loading of charts
ScrollBars = ssBoth TabOrder = 0	from www URL''s'
end	II VIII VIIII VIIII S
<> object Panel2: TPanel	'** JPEG, PNG, GIF, PCX,
Left = 0	Metafile *.WMF and Bitmap *.BMP
Top = 0	file and' +
Width = 467	' clipboard support'
Height = 60	'** Accepts any child control
Align = alTop	(TeeChart derives from TPanel).'
TabOrder = 1	'** No VBX''s or OCX''s
<> object Label1: TLabel	required. Just packages and units!'
Left = 164	'** No drawing flickering.'
Top = 35	'** Mini-charts capable'
Width = 160	'** Many Example Projects and
Height = 16	a Tips Database'
HelpType = htKeyword	
Caption = 'Product	'** Lines, Scatter, Area,
Specifications.'	Vertical and Horizontal Bars,

more than 250 new features over the
'standard version included in
Delphi and C++ Builder.'
***
'Visit us at
http://www.steema.com and
download the evaluation ve' +
'rsion.'
")
TabOrder = 0
end
end
End
16
<> object ColoredForm:
TColoredForm
Left = 189
Top = 111
Width = 574
Height = 439
Caption = 'TeeChart Custom
Coloring Example'
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Chart1: TChart
Left = 137
Top = 0
Height = 405
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
Foot.Font.Charset =
DEFAULT_CHARSET
Foot.Font.Color = clRed
Foot.Font.Height = -12
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsItalic]
Title.Alignment = taRightJustify
Title.Text.Strings = (
'Each Point has it''s own Color')
Chart3DPercent = 10

Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
end
<> object Image2: TImage
Left = 14
Top = 8
Width = 141
Height = 45
HelpType = htKeyword
AutoSize = True
Picture.Data = {
end
<> object BitBtn1: TBitBtn
Left = 368
Top = 20
Width = 89
Height = 33
TabOrder = 0
Kind = bkClose
end
end
<> object Panel1: TPanel
Left = 0
Top = 60
Width = 467
Height = 107
Align = alTop
TabOrder = 2
<> object Memo2: TMemo
Left = 7
Top = 5
Width = 450
Height = 92
Lines.Strings = (
,
'TeeChart Pro 5.0 is a library
of 100% Native Borland Delphi
char' +
'ting '
'and graphing components
ready to include in your
applications.'
'TeeChart Pro 5.0 includes
- II I II I I I I I I I I I I I I

Height = 405
Align = alLeft
TabOrder = 1
<> object CheckBox1:
TCheckBox
Left = 7
Top = 20
Width = 114
Height = 17
Caption = 'Left Axis Inverted'
Checked = True
State = cbChecked
TabOrder = 0
OnClick = CheckBox1Click
end
<> object BitBtn2: TBitBtn
Left = 19
Top = 316
Width = 89
Height = 33
TabOrder = 1
Kind = bkClose
end
<> object Memo1: TMemo
Left = 8
Top = 72
Width = 121
Height = 161
Lines.Strings = (
'This demo shows how '
'to set a different color '
'for each point after'
'points have been added '
'to the Series.'
"
'It shows too how to '
'invert Axis using the axis'
'Inverted property.')
TabOrder = 2
end
end
End
111111111111111111111111111111111111111
17
<> object CustomAxisForm:
TCustomAxisForm
Left = 166
Top = 110

LeftAxis.Inverted = True
Legend.Alignment = laBottom
Align = alClient
TabOrder = 0
<> object LineSeries1:
TLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clRed
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
Stairs = True
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
end
<> object PointSeries1:
<b>TPointSeries</b>
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clGreen
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = True
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
end
end
<> object Panel1: TPanel
Service and the service of the servi
Left = 0

Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clRed
AfterDrawValues =
LineSeries1AfterDrawValues
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 101
Top = 63
end
<> object PointSeries1:
<b>TPointSeries</b>
ColorEachPoint = True
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clGreen
Pointer.InflateMargins = True
Pointer.Style = psDiamond
Pointer.Visible = True
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 147
Top = 61
end
<> object FastLineSeries1:
TFastLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clBlue
LinePen.Color = clBlue

****
Width = 603
Height = 402
Caption = 'TeeChart Custom Axis
Drawing Example'
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Chart1: TChart
Left = 137
Top = 0
Width = 458
Height = 368
AnimatedZoom = True
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
Foot.Font.Charset =
DEFAULT CHARSET
Foot.Font.Color = clRed
Foot.Font.Height = -12
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsItalic]
Foot.Frame.Color = clScrollBar
Gradient.Visible = True
Title.Frame.Color = clScrollBar
Title.Text.Strings = (
'TChart Component')
Title.Visible = False
BottomAxis.Grid.Color =
clScrollBar
BottomAxis.Grid.Visible = False
LeftAxis.Grid.Color = clScrollBar
LeftAxis.Grid.Visible = False
Legend.Alignment = laLeft
RightAxis.Grid.Color =
clScrollBar
RightAxis.Grid.Visible = False
TopAxis.Grid.Color = clScrollBar
TopAxis.Grid.Visible = False
View3D = False
View3DWalls = False
Align = alClient
TabOrder = 0
<> object LineSeries1:
TLineSeries

Left = 9
Top = 92
Width = 97
Height = 17
Caption = 'Invert Axis '
TabOrder = 2
OnClick = CheckBox2Click
end
<> object DrawGrid: TCheckBox
Left = 9
Top = 60
Width = 80
Height = 17
Caption = 'Draw Grid'
Checked = True
State = cbChecked
TabOrder = 3
OnClick = DrawGridClick
end
<> object Memo1: TMemo
Left = 8
Top = 136 Width = 121
Height = 153
Lines.Strings = (
'This example shows '
'how to draw custom '
'Axis at specific '
'positions.'
"
'The Axis CustomDraw'
'method is use to display '
'custom Axis.')
TabOrder = 4
end
end
<> object Timer1: TTimer
Enabled = False
Interval = 1
OnTimer = Timer1Timer
Left = 31
Top = 34
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
18
<> object CrossHairForm:

XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.0000000000000000000
YValues.Order = loNone
Left = 187
Top = 64
end
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 137
Height = 368
Align = alLeft
TabOrder = 1
<> object CheckBox1:
TCheckBox
Left = 9
Top = 12
Width = 97
Height = 17
Caption = '&Animate !!!'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -16
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 0
OnClick = CheckBox1Click
end
<> object BitBtn1: TBitBtn
Left = 25
Top = 324
Width = 80
Height = 33
TabOrder = 1
Kind = bkClose
end
<> object CheckBox2:
TCheckBox

Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.0000000000000000000
YValues.Order = loNone
Left = 199
Top = 121
end
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 145
Height = 370
Align = alLeft
TabOrder = 1
<> object Label1: TLabel
Left = 44
Top = 12
Width = 15
Height = 13
HelpType = htKeyword
Caption = '0,0'
end
<> object Label2: TLabel
Left = 16
Top = 12
Width = 20
Height = 13
HelpType = htKeyword
Caption = 'X,Y:'
end
<> object BitBtn1: TBitBtn
Left = 26
Top = 324
Width = 89
Height = 33
TabOrder = 0
Kind = bkClose
end
<> object CheckBox1:
object Checkboat.

TCrossHairForm
Left = 200
Top = 111
Width = 506
Height = 404
ActiveControl = Chart1
Caption = 'TeeChart Cross Hair
Demo'
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Chart1: TChart
Left = 145
Top = 0
Width = 353
Height = 370
Cursor = crCross
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
Foot.Font.Charset =
DEFAULT CHARSET
Foot.Font.Color = clRed
Foot.Font.Height = -12
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsItalic]
Title.Text.Strings = (
'TChart Component')
Title.Visible = False
BottomAxis.Grid.Visible = False
Chart3DPercent = 30
LeftAxis.Grid.Visible = False
View3D = False
Align = alClient
TabOrder = 0
OnMouseMove =
Chart1MouseMove
<> object LineSeries1:
TLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clRed
AfterDrawValues =
LineSeries1AfterDrawValues
Pointer.InflateMargins = True
1 vinter anniately and gills 11 ut

Caption = 'TeeChart Example.
Horizontal and Vertical Bars with
DBChart'
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
OnShow = FormShow
PixelsPerInch = 96
TextHeight = 13
<> object DBChart1:
TDBChart
Left = 145
Top = 0
Width = 455
Height = 439
AnimatedZoom = True
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Color = clWhite
BottomWall.Color = 16777088
Foot.Brush.Color = clBlue
Foot.Color = clBlue
Foot.Font.Charset =
DEFAULT CHARSET
Foot.Font.Color = clLime
Foot.Font.Height = -13
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsBold, fsItalic]
Foot.Frame.Color = clWhite
Foot.Frame.Width = 5
Foot.Frame.Visible = True
Foot.Text.Strings = (
'Who has all of them ?')
,
Title.Alignment = taLeftJustify Title.Brush.Color = clNavy
Title.Color = clNavy
Title.Font.Charset =
DEFAULT_CHARSET
Title.Font.Color = clYellow
Title.Font.Height = -19
Title.Font.Name = 'Arial'
Title.Font.Style = [fsBold, fsItalic]
Title.Frame.Color = clBlue
Title.Frame.Width = 5
Title.Frame.Visible = True
Title.Text.Strings = (
'Animal Comparison'
·

TCheckBox
Left = 15
Top = 44
<b>Width</b> = 97
Height = 17
Caption = 'C&ross Cursor'
Checked = True
State = cbChecked
TabOrder = 1
OnClick = CheckBox1Click
end
<> object BitBtn2: TBitBtn
Left = 11
Top = 260
Width = 118
Height = 33
Caption = 'CrossHair C&olor'
TabOrder = 2
OnClick = BitBtn2Click
end
<> object Memo1: TMemo
Left = 8
Top = 80
Width = 121
Height = 161
Lines.Strings = (
'Cross-hair lines are '
'drawn over a TChart '
'using the '
'OnMouseMove event.'
11
'Coordinates for the '
'cross-hair lines are '
'calculated using the '
'Series GetCursorValues '
'method.')
TabOrder = 3
end
end
End
19
<> object DBHorizBarForm:
TDBHorizBarForm
Left = 185
Top = 109
Width = 608
Height = 473

DEFAULT_CHARSET	'Size vs. Weight')
TopAxis.LabelsFont.Color =	BackColor = clWhite
clRed	BottomAxis.LabelsFont.Charset =
TopAxis.LabelsFont.Height = -13	DEFAULT_CHARSET
TopAxis.LabelsFont.Name =	BottomAxis.LabelsFont.Color =
'Arial'	clGreen
TopAxis.LabelsFont.Style =	BottomAxis.LabelsFont.Height = -
[fsItalic]	13
TopAxis.Title.Caption = 'Weight	BottomAxis.LabelsFont.Name =
in Kg'	'Arial'
TopAxis.Title.Font.Charset =	BottomAxis.LabelsFont.Style =
DEFAULT_CHARSET	[fsBold, fsItalic]
TopAxis.Title.Font.Color = clBlue	BottomAxis.LabelStyle = talValue
TopAxis.Title.Font.Height = -16	<b>BottomAxis.Title.Caption = 'Size</b>
TopAxis.Title.Font.Name = 'Arial'	in Inches'
TopAxis.Title.Font.Style = []	BottomAxis.Title.Font.Charset =
Align = alClient	DEFAULT_CHARSET
TabOrder = 0	BottomAxis.Title.Font.Color =
<> object HorizBarSeries1:	clYellow
THorizBarSeries	BottomAxis.Title.Font.Height = -
HorizAxis = aTopAxis	16
Marks.Arrow.Color = clLime	BottomAxis.Title.Font.Name =
Marks.ArrowLength = 20	'Arial'
Marks.BackColor = clWhite	BottomAxis.Title.Font.Style =
Marks.Font.Charset =	[fsBold]
DEFAULT_CHARSET	Chart3DPercent = 50
Marks.Font.Color = clRed	LeftAxis.Inverted = True
Marks.Font.Height = -13	LeftAxis.LabelsFont.Charset =
Marks.Font.Name = 'Arial'	DEFAULT_CHARSET
Marks.Font.Style = [fsBold]	LeftAxis.LabelsFont.Color =
Marks.Style = smsXValue	clNavy
Marks.Visible = True	LeftAxis.LabelsFont.Height = -13
DataSource = Table1	LeftAxis.LabelsFont.Name =
SeriesColor = clRed	'Arial'
Title = 'Weight'	LeftAxis.LabelsFont.Style =
XLabelsSource = 'NAME'	[fsBold]
MultiBar = mbNone	LeftAxis.LabelsSeparation = 5
XValues.DateTime = False	LeftAxis.LabelStyle = talText
XValues.Name = 'Bar'	LeftAxis.Title.Caption = 'Animal
XValues.Multiplier =	Name'
1.000000000000000000	LeftAxis.Title.Font.Charset =
XValues.Order = loNone	DEFAULT_CHARSET
XValues.ValueSource =	LeftAxis.Title.Font.Color = clRed
'WEIGHT'	LeftAxis.Title.Font.Height = -19
YValues.DateTime = False	LeftAxis.Title.Font.Name = 'Arial'
YValues.Name = 'Y'	LeftAxis.Title.Font.Style =
YValues.Multiplier =	[fsBold, fsItalic]
1.000000000000000000	Legend.ColorWidth = 60
YValues.Order = loNone	TopAxis.LabelsFont.Charset =
	ı

Grand Louis I. I.
State = cbChecked
TabOrder = 0
OnClick = CheckBox1Click
end
<> object RadioGroup1:
TRadioGroup
Left = 6
<b>Top</b> = 44
Width = 123
Height = 101
Caption = 'MultiBar Style'
ItemIndex = 0
Items.Strings = (
'&None'
'S&ide to Side'
'&Stacked'
'S&tacked 100%')
TabOrder = 1
OnClick = RadioGroup1Click
end
<> object BitBtn2: TBitBtn
Left = 22
Top = 400
Width = 95
Height = 35
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 2
Kind = bkClose
end
<> object BitBtn3: TBitBtn
Left = 8
Top = 359
Width = 121
Height = 34
Caption = 'Swap Series'
TabOrder = 3
OnClick = BitBtn3Click
Glyph.Data = {
0303}
NumGlyphs = 2
end
<> object CheckBox2:
<pre>&lt;&gt; onlect UneckBox2:</pre>

end
<> object HorizBarSeries2:
THorizBarSeries
Marks.Arrow.Color = clYellow
Marks.ArrowLength = 20
Marks.BackColor = clWhite
Marks.Font.Charset =
DEFAULT CHARSET
Marks.Font.Color = clGreen
Marks.Font.Height = -13
Marks.Font.Name = 'Arial'
Marks.Font.Style = [fsBold,
fsItalic
Marks.Style = smsXValue Marks.Visible = True
DataSource = Table1
SeriesColor = clBlue
Title = 'Size'
XLabelsSource = 'NAME'
MultiBar = mbNone
XValues.DateTime = False
XValues.Name = 'Bar'
XValues.Multiplier =
1.00000000000000000000
XValues.Order = loNone
XValues.ValueSource = 'SIZE'
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
end
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 145
Height = 439
Align = alLeft
TabOrder = 1
<> object CheckBox1:
TCheckBox
Left = 42
Top = 155
Width = 47
Height = 17
Caption = '3&D'
Checked = True
CHUCKU - IIUC

Top = 17 end End	Y 6: 202
end End	Left = 382
End	•
<pre>Compose to the state of th</pre>	
<pre>c&gt; object DemoForm: TDemoForm Left = 189 Top = 110 Width = 628 Height = 440 ActiveControl = ListBox1 Caption = 'DemoForm' Color = clNavy Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate OnShow = FormShow PixelsPerInch = 96 TextHeight = 13 &lt;&gt; object Label4: TLabel Left = 121 Top = 260 Width = 23 Height = 13 HelpType = htKeyword Alignment = taRightJustify Caption = 'Slow' end &lt;&gt; object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end &lt;&gt; object Panel2: TPanel Left = 0 Top = 367</pre>	
C> object DemoForm:  TDemoForm  Left = 189  Top = 110  Width = 628  Height = 440  ActiveControl = ListBox1  Caption = 'DemoForm'  Color = clNavy  Font.Charset =  DEFAULT CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  OldCreateOrder = True  Position = poScreenCenter  OnCreate = FormCreate  OnShow = FormShow  PixelsPerInch = 96  TextHeight = 13  <> object Label4: TLabel  Left = 121  Top = 260  Width = 23  Height = 13  HelpType = htKeyword  Alignment = taRightJustify  Caption = 'Slow' end  <> object Label5: TLabel  Left = 12  Top = 260  Width = 20  Height = 13  HelpType = htKeyword  Caption = 'Fast' end  <> object Panel2: TPanel  Left = 0  Top = 367	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
TDemoForm  Left = 189  Top = 110  Width = 628  Height = 440  ActiveControl = ListBox1  Caption = 'DemoForm'  Color = clNavy  Font.Charset =  DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  OldCreateOrder = True  Position = poScreenCenter  OnCreate = FormCreate  OnShow = FormShow  PixelsPerInch = 96  TextHeight = 13  <> object Label4: TLabel  Left = 121  Top = 260  Width = 23  Height = 13  HelpType = htKeyword  Alignment = taRightJustify  Caption = 'Slow' end  <> object Label5: TLabel  Left = 12  Top = 260  Width = 20  Height = 13  HelpType = htKeyword  Caption = 'Fast' end  <> object Panel2: TPanel  Left = 0  Top = 367	20
Left = 189  Top = 110  Width = 628  Height = 440  ActiveControl = ListBox1  Caption = 'DemoForm'  Color = clNavy  Font.Charset =  DEFAULT CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  OldCreateOrder = True  Position = poScreenCenter  OnCreate = FormCreate  OnShow = FormShow  PixelsPerInch = 96  TextHeight = 13  <> object Label4: TLabel  Left = 121  Top = 260  Width = 23  Height = 13  HelpType = htKeyword  Alignment = taRightJustify  Caption = 'Slow'  end  <> object Label5: TLabel  Left = 12  Top = 260  Width = 20  Height = 13  HelpType = htKeyword  Caption = 'Fast'  end  <> object Panel2: TPanel  Left = 0  Top = 367	<> object DemoForm:
Top = 110  Width = 628  Height = 440  ActiveControl = ListBox1  Caption = 'DemoForm'  Color = clNavy  Font.Charset =  DEFAULT CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  OldCreateOrder = True  Position = poScreenCenter  OnCreate = FormCreate  OnShow = FormShow  PixelsPerInch = 96  TextHeight = 13  <> object Label4: TLabel  Left = 121  Top = 260  Width = 23  Height = 13  HelpType = htKeyword  Alignment = taRightJustify  Caption = 'Slow'  end  <> object Label5: TLabel  Left = 12  Top = 260  Width = 20  Height = 13  HelpType = htKeyword  Caption = 'Fast'  end  <> object Panel2: TPanel  Left = 0  Top = 367	
Width = 628 Height = 440 ActiveControl = ListBox1 Caption = 'DemoForm' Color = clNavy Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate OnShow = FormShow PixelsPerInch = 96 TextHeight = 13 <> object Label4: TLabel Left = 121 Top = 260 Width = 23 Height = 13 HelpType = htKeyword Alignment = taRightJustify Caption = 'Slow' end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	Left = 189
Height = 440 ActiveControl = ListBox1 Caption = 'DemoForm' Color = clNavy Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate OnShow = FormShow PixelsPerInch = 96 TextHeight = 13 <> object Label4: TLabel Left = 121 Top = 260 Width = 23 Height = 13 HelpType = htKeyword Alignment = taRightJustify Caption = 'Slow' end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	Top = 110
ActiveControl = ListBox1  Caption = 'DemoForm'  Color = clNavy  Font.Charset =  DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  OldCreateOrder = True  Position = poScreenCenter  OnCreate = FormCreate  OnShow = FormShow  PixelsPerInch = 96  TextHeight = 13  <> object Label4: TLabel  Left = 121  Top = 260  Width = 23  Height = 13  HelpType = htKeyword  Alignment = taRightJustify  Caption = 'Slow'  end  <> object Label5: TLabel  Left = 12  Top = 260  Width = 20  Height = 13  HelpType = htKeyword  Caption = 'Fast'  end  <> object Panel2: TPanel  Left = 0  Top = 367	Width = 628
Caption = 'DemoForm'  Color = clNavy  Font.Charset =  DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  OldCreateOrder = True  Position = poScreenCenter  OnCreate = FormCreate  OnShow = FormShow  PixelsPerInch = 96  TextHeight = 13  <> object Label4: TLabel  Left = 121  Top = 260  Width = 23  Height = 13  HelpType = htKeyword  Alignment = taRightJustify  Caption = 'Slow'  end  <> object Label5: TLabel  Left = 12  Top = 260  Width = 20  Height = 13  HelpType = htKeyword  Caption = 'Fast'  end  <> object Panel2: TPanel  Left = 0  Top = 367	Height = 440
Color = clNavy Font.Charset = DEFAULT_CHARSET Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate OnShow = FormShow PixelsPerInch = 96 TextHeight = 13 <> object Label4: TLabel Left = 121 Top = 260 Width = 23 Height = 13 HelpType = htKeyword Alignment = taRightJustify Caption = 'Slow' end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	ActiveControl = ListBox1
Font.Charset = DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = [] OldCreateOrder = True  Position = poScreenCenter  OnCreate = FormCreate  OnShow = FormShow  PixelsPerInch = 96  TextHeight = 13  <> object Label4: TLabel  Left = 121  Top = 260  Width = 23  Height = 13  HelpType = htKeyword  Alignment = taRightJustify  Caption = 'Slow' end  <> object Label5: TLabel  Left = 12  Top = 260  Width = 20  Height = 13  HelpType = htKeyword  Caption = 'Fast' end  <> object Panel2: TPanel  Left = 0  Top = 367	Caption = 'DemoForm'
Font.Charset = DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = [] OldCreateOrder = True  Position = poScreenCenter  OnCreate = FormCreate  OnShow = FormShow  PixelsPerInch = 96  TextHeight = 13  <> object Label4: TLabel  Left = 121  Top = 260  Width = 23  Height = 13  HelpType = htKeyword  Alignment = taRightJustify  Caption = 'Slow' end  <> object Label5: TLabel  Left = 12  Top = 260  Width = 20  Height = 13  HelpType = htKeyword  Caption = 'Fast' end  <> object Panel2: TPanel  Left = 0  Top = 367	Color = clNavy
DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -11  Font.Name = 'MS Sans Serif'  Font.Style = []  OldCreateOrder = True  Position = poScreenCenter  OnCreate = FormCreate  OnShow = FormShow  PixelsPerInch = 96  TextHeight = 13  <> object Label4: TLabel  Left = 121  Top = 260  Width = 23  Height = 13  HelpType = htKeyword  Alignment = taRightJustify  Caption = 'Slow'  end  <> object Label5: TLabel  Left = 12  Top = 260  Width = 20  Height = 13  HelpType = htKeyword  Caption = 'Fast'  end  <> object Panel2: TPanel  Left = 0  Top = 367	· ·
Font.Color = clBlack Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate OnShow = FormShow PixelsPerInch = 96 TextHeight = 13 <> object Label4: TLabel Left = 121 Top = 260 Width = 23 Height = 13 HelpType = htKeyword Alignment = taRightJustify Caption = 'Slow' end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
Font.Height = -11 Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate OnShow = FormShow PixelsPerInch = 96 TextHeight = 13 <> object Label4: TLabel Left = 121 Top = 260 Width = 23 Height = 13 HelpType = htKeyword Alignment = taRightJustify Caption = 'Slow' end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
Font.Name = 'MS Sans Serif' Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate OnShow = FormShow PixelsPerInch = 96 TextHeight = 13 <> object Label4: TLabel Left = 121 Top = 260 Width = 23 Height = 13 HelpType = htKeyword Alignment = taRightJustify Caption = 'Slow' end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
Font.Style = [] OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate OnShow = FormShow PixelsPerInch = 96 TextHeight = 13 <> object Label4: TLabel Left = 121 Top = 260 Width = 23 Height = 13 HelpType = htKeyword Alignment = taRightJustify Caption = 'Slow' end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate OnShow = FormShow PixelsPerInch = 96 TextHeight = 13 <> object Label4: TLabel Left = 121 Top = 260 Width = 23 Height = 13 HelpType = htKeyword Alignment = taRightJustify Caption = 'Slow' end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
Position = poScreenCenter OnCreate = FormCreate OnShow = FormShow PixelsPerInch = 96  TextHeight = 13 <> object Label4: TLabel Left = 121 Top = 260 Width = 23 Height = 13 HelpType = htKeyword Alignment = taRightJustify Caption = 'Slow' end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
OnCreate = FormCreate OnShow = FormShow PixelsPerInch = 96  TextHeight = 13 <> object Label4: TLabel Left = 121  Top = 260 Width = 23 Height = 13 HelpType = htKeyword Alignment = taRightJustify Caption = 'Slow' end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
OnShow = FormShow  PixelsPerInch = 96  TextHeight = 13  <> object Label4: TLabel  Left = 121  Top = 260  Width = 23  Height = 13  HelpType = htKeyword  Alignment = taRightJustify  Caption = 'Slow' end  <> object Label5: TLabel  Left = 12  Top = 260  Width = 20  Height = 13  HelpType = htKeyword  Caption = 'Fast' end  <> object Panel2: TPanel  Left = 0  Top = 367	-
PixelsPerInch = 96 TextHeight = 13 <> object Label4: TLabel Left = 121 Top = 260 Width = 23 Height = 13 HelpType = htKeyword Alignment = taRightJustify Caption = 'Slow' end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
TextHeight = 13  <> object Label4: TLabel  Left = 121  Top = 260  Width = 23  Height = 13  HelpType = htKeyword  Alignment = taRightJustify  Caption = 'Slow' end  <> object Label5: TLabel  Left = 12  Top = 260  Width = 20  Height = 13  HelpType = htKeyword  Caption = 'Fast' end  <> object Panel2: TPanel  Left = 0  Top = 367	
<> object Label4: TLabel Left = 121 Top = 260 Width = 23 Height = 13 HelpType = htKeyword Alignment = taRightJustify Caption = 'Slow' end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
Left = 121 Top = 260 Width = 23 Height = 13 HelpType = htKeyword Alignment = taRightJustify Caption = 'Slow' end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
Top = 260 Width = 23 Height = 13 HelpType = htKeyword Alignment = taRightJustify Caption = 'Slow' end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
Width = 23 Height = 13 HelpType = htKeyword Alignment = taRightJustify Caption = 'Slow' end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
Height = 13 HelpType = htKeyword Alignment = taRightJustify Caption = 'Slow' end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
HelpType = htKeyword Alignment = taRightJustify Caption = 'Slow' end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
Alignment = taRightJustify Caption = 'Slow' end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
Caption = 'Slow' end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	1 01
end <> object Label5: TLabel Left = 12 Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
<> object Label5: TLabel  Left = 12  Top = 260  Width = 20  Height = 13  HelpType = htKeyword  Caption = 'Fast' end  <> object Panel2: TPanel  Left = 0  Top = 367	
Left = 12  Top = 260  Width = 20  Height = 13  HelpType = htKeyword  Caption = 'Fast' end  <> object Panel2: TPanel  Left = 0  Top = 367	
Top = 260 Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
Width = 20 Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
Height = 13 HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
HelpType = htKeyword Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	
Caption = 'Fast' end <> object Panel2: TPanel Left = 0 Top = 367	Č
end <> object Panel2: TPanel Left = 0 Top = 367	
<> object Panel2: TPanel  Left = 0  Top = 367	•
Left = 0 Top = 367	
Top = 367	<> object Panel2: TPanel
Width = $62\overline{0}$	Top = 367
	Width = 620

TCheckBox
Left = 8
Top = 8
Width = 137
Height = 25
Caption = '&Animate !!!'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -19
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 4
OnClick = CheckBox2Click
end
<> object Memo1: TMemo
Left = 8
Top = 184
Width = 121
Height = 161
Lines.Strings = (
'Connecting a Series to '
'a TTable component is '
'just at few mouse clicks '
'using the Chart editor '
'dialog!'
ulalog:
Un this arounds 2 fields !
'In this example 2 fields ' 'from ANIMALS.DBF '
'table are connected to '
'2 Horizontal Bar'
'Series.')
TabOrder = 5
end
end
<> object Table1: TTable
Active = True
DatabaseName = 'DBDEMOS'
ReadOnly = True
TableName = 'ANIMALS.DBF'
Left = 431
Top = 18
end
<> object Timer1: TTimer
Enabled = False
Interval = 1
OnTimer = Timer1Timer

Width = 85
Height = 25
Caption = 'Zoom Out'
TabOrder = 2
OnClick = BitBtn2Click
Glyph.Data = {
33333773FF77333333333370007333
33333333777333333333}
NumGlyphs = 2
end
<> object Button3: TButton
Left = 204
Top = 7
Width = 75
Height = 25
Caption = 'Reset Zoom'
Enabled = False
TabOrder = 3
OnClick = Button3Click
end
end
<> object Panel3: TPanel
Left = 453
Top = 0
Width = 167
Height = 367
Align = alRight
TabOrder = 2
<> object Image1: TImage
Left = 14
Top = 6
Width = 141
Height = 45
HelpType = htKeyword
AutoSize = True
Picture.Data = {
07544269746D6170861D0000424D86
1D00000000000036040000280000008
D00
00002D0000000100080000000000501
9000000000000000000000001000000
01
0000000000008080800000008000008
0800000800000808000008000000080
00

Height = 39
Align = alBottom
TabOrder = 1
<> object SpeedButton1:
TSpeedButton
Left = 312
Top = 7
Width = 25
Height = 25
HelpType = htKeyword
Glyph.Data = {
NumGlyphs = 2
OnClick = SpeedButton1Click
end
<> object SpeedButton2:
TSpeedButton
Left = 344
Top = 7
Width = 25
Height = 25
HelpType = htKeyword
Glyph.Data = {
NumGlyphs = 2
OnClick = SpeedButton4Click
end
<> object Button2: TButton
Left = 492
Top = 7
Width = 75
Height = 25
8
Caption = '< Back'
TabOrder = 0
OnClick = Button2Click
end
<> object BitBtn1: TBitBtn
Left = 16
<b>Top</b> = 7
Width = 81
Height = 25
Caption = 'Zoom In'
TabOrder = 1
OnClick = BitBtn1Click
Glyph.Data = {
NumGlyphs = 2
end
<> object BitBtn2: TBitBtn
Left = 110
<b>Top</b> = 7

W/: J4b = 42
Width = 43
Height = 13
HelpType = htKeyword
Caption = '&Rotation:'
FocusControl = ScrollBar3
end
<> object ListBox1:
TListBox
Left = 12
Top = 82
Width = 145
Height = 152
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ItemHeight = 13
Items.Strings = (
'Line / Strip'
'Bar / Stacked'
'Horizontal Bar /Stacked'
'Area / Step'
'Point XY / Scatter'
'Pie'
'Fast Line'
'Arrow'
'Bubble'
'Gantt'
'Shapes')
ParentFont = False
TabOrder = 0
OnClick = ListBox1Click
end
<> object CheckBox1:
TCheckBox
Left = 53
Top = 292
Width = 37
Height = 17
Caption = '3&D:'
Checked = True
State = cbChecked
TabOrder = 1
OnClick = CheckBox1Click
end
<> object CheckBox2:

0.001
OnClick = Image1Click
end
<> object Label1: TLabel
Left = 12
Top = 66
Width = 64
Height = 13
HelpType = htKeyword
Caption = '&Series Types:'
FocusControl = ListBox1
end
<> object Label2: TLabel
Left = 17
Top = 247
Width = 20
Height = 13
HelpType = htKeyword
Caption = 'Fast'
end
<> object Label3: TLabel
Left = 126
Top = 247
Width = 23
Height = 13
HelpType = htKeyword
Alignment = taRightJustify
Caption = 'Slow'
end
<> object Label6: TLabel
Left = 17
Top = 300
Width = 14
Height = 13
HelpType = htKeyword
Caption = '2D'
end
<> object Label7: TLabel
Left = 135
Top = 300
Width = 14
Height = 13
HelpType = htKeyword
Alignment = taRightJustify
Caption = '3D'
1
end
<> object Label8: TLabel
Left = 8
Top = 346

TabOrder = 5
OnChange = ScrollBar3Change
end
end
<> object Notebook1:
TNotebook
Left = 0
Top = 0
Width = 453
Height = 367
Align = alClient
TabOrder = 0
<> object TPage
Left = 0
Top = 0
Caption = 'Line'
<> object Chart1: TChart
Left = 0
Top = 0
Width = 453
Height = 367
BackWall.Brush.Color =
clWhite
BackWall.Brush.Style =
bsClear
BackWall.Color = clWhite
BottomWall.Color = 16777088
BottomWall.Size = 9
LeftWall.Color = 8454016
LeftWall.Size = 9
Title.Brush.Color = 8388863
Title.Color = 8388863
Title.Font.Charset =
DEFAULT_CHARSET
Title.Font.Color = clWhite
Title.Font.Height = -19
Title.Font.Name = 'Arial'
Title.Font.Style = [fsBold,
fsItalic]
Title.Frame.Width = 3
Title.Frame.Visible = True
Title.Text.Strings = (
'Worldwide Fuel
Consumption'
'( in millions of liters )')
BackColor = clWhite
BottomAxis.Axis.Color = clBlue
BottomAxis.Grid.Visible =
False
1 4130

TCheckBox
Left = 53
Top = 239
Width = 57
Height = 17
Caption = '&Animate'
Checked = True
State = cbChecked
TabOrder = 2
OnClick = CheckBox2Click
end
<> object ScrollBar1:
TScrollBar
Left = 17
Top = 263
Width = 133
Height = 16
LargeChange = 100
Max = 1000
Min = 1
PageSize = 0
Position = 1
SmallChange = 25
TabOrder = 3
OnChange = ScrollBar1Change
end
<> object ScrollBar2:
TScrollBar
Left = 17
Top = 316
Width = 133
Height = 16 Min = 1
PageSize = 0
Position = 15
Position = 15 TabOrder = 4
Position = 15 TabOrder = 4 OnChange = ScrollBar2Change
Position = 15 TabOrder = 4 OnChange = ScrollBar2Change end
Position = 15 TabOrder = 4 OnChange = ScrollBar2Change end <> object ScrollBar3:
Position = 15 TabOrder = 4 OnChange = ScrollBar2Change end <> object ScrollBar3: TScrollBar
Position = 15 TabOrder = 4 OnChange = ScrollBar2Change end <> object ScrollBar3: TScrollBar Left = 56
Position = 15 TabOrder = 4 OnChange = ScrollBar2Change end <> object ScrollBar3: TScrollBar Left = 56 Top = 344
Position = 15 TabOrder = 4 OnChange = ScrollBar2Change end <> object ScrollBar3: TScrollBar Left = 56 Top = 344 Width = 94
Position = 15 TabOrder = 4 OnChange = ScrollBar2Change end <> object ScrollBar3: TScrollBar Left = 56 Top = 344 Width = 94 Height = 16
Position = 15 TabOrder = 4 OnChange = ScrollBar2Change end <> object ScrollBar3: TScrollBar Left = 56 Top = 344 Width = 94
Position = 15 TabOrder = 4 OnChange = ScrollBar2Change end <> object ScrollBar3: TScrollBar Left = 56 Top = 344 Width = 94 Height = 16
Position = 15 TabOrder = 4 OnChange = ScrollBar2Change end <> object ScrollBar3: TScrollBar Left = 56 Top = 344 Width = 94 Height = 16 Max = 360 Min = 270
Position = 15 TabOrder = 4 OnChange = ScrollBar2Change end <> object ScrollBar3: TScrollBar Left = 56 Top = 344 Width = 94 Height = 16 Max = 360

# <u>AC.&P</u>

T LE (C) L (CD LII
Legend.Font.Style = [fsBold]
Align = alClient
Color = 8454143
TabOrder = 0
<> object Series1:
TLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = 65408
Title = 'Europe'
Pointer.Brush.Color = clRed
Pointer.HorizSize = 6
Pointer.InflateMargins = False
Pointer.Style = psCircle
Pointer.VertSize = 6
Pointer.Visible = True
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
end
<> object Series2:
TLineSeries
Marks ArrowLength = 8
Marks.ArrowLength = 8
Marks.Visible = False
Marks.Visible = False SeriesColor = clAqua
Marks.Visible = False SeriesColor = clAqua Title = 'America'
Marks.Visible = False SeriesColor = clAqua Title = 'America' Pointer.Brush.Color = clLime
Marks.Visible = False SeriesColor = clAqua Title = 'America' Pointer.Brush.Color = clLime Pointer.HorizSize = 6
Marks.Visible = False SeriesColor = clAqua Title = 'America' Pointer.Brush.Color = clLime Pointer.HorizSize = 6 Pointer.InflateMargins = False
Marks.Visible = False SeriesColor = clAqua Title = 'America' Pointer.Brush.Color = clLime Pointer.HorizSize = 6 Pointer.InflateMargins = False Pointer.Style = psTriangle
Marks.Visible = False SeriesColor = clAqua Title = 'America' Pointer.Brush.Color = clLime Pointer.HorizSize = 6 Pointer.InflateMargins = False Pointer.Style = psTriangle Pointer.VertSize = 6
Marks.Visible = False SeriesColor = clAqua Title = 'America' Pointer.Brush.Color = clLime Pointer.HorizSize = 6 Pointer.InflateMargins = False Pointer.Style = psTriangle
Marks.Visible = False SeriesColor = clAqua Title = 'America' Pointer.Brush.Color = clLime Pointer.HorizSize = 6 Pointer.InflateMargins = False Pointer.Style = psTriangle Pointer.VertSize = 6
Marks.Visible = False SeriesColor = clAqua Title = 'America' Pointer.Brush.Color = clLime Pointer.HorizSize = 6 Pointer.InflateMargins = False Pointer.Style = psTriangle Pointer.VertSize = 6 Pointer.Visible = True
Marks.Visible = False SeriesColor = clAqua Title = 'America' Pointer.Brush.Color = clLime Pointer.HorizSize = 6 Pointer.InflateMargins = False Pointer.Style = psTriangle Pointer.VertSize = 6 Pointer.Visible = True XValues.DateTime = False XValues.Name = 'X'
Marks.Visible = False SeriesColor = clAqua Title = 'America' Pointer.Brush.Color = clLime Pointer.HorizSize = 6 Pointer.InflateMargins = False Pointer.Style = psTriangle Pointer.VertSize = 6 Pointer.Visible = True XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier =
Marks.Visible = False SeriesColor = clAqua Title = 'America' Pointer.Brush.Color = clLime Pointer.HorizSize = 6 Pointer.InflateMargins = False Pointer.Style = psTriangle Pointer.VertSize = 6 Pointer.Visible = True XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000
Marks.Visible = False SeriesColor = clAqua Title = 'America' Pointer.Brush.Color = clLime Pointer.HorizSize = 6 Pointer.InflateMargins = False Pointer.Style = psTriangle Pointer.VertSize = 6 Pointer.Visible = True XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000
Marks.Visible = False SeriesColor = clAqua Title = 'America' Pointer.Brush.Color = clLime Pointer.HorizSize = 6 Pointer.InflateMargins = False Pointer.Style = psTriangle Pointer.VertSize = 6 Pointer.Visible = True XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000
Marks.Visible = False SeriesColor = clAqua Title = 'America' Pointer.Brush.Color = clLime Pointer.HorizSize = 6 Pointer.InflateMargins = False Pointer.Style = psTriangle Pointer.VertSize = 6 Pointer.Visible = True XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.00000000000000000 XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y'
Marks.Visible = False SeriesColor = clAqua Title = 'America' Pointer.Brush.Color = clLime Pointer.HorizSize = 6 Pointer.InflateMargins = False Pointer.Style = psTriangle Pointer.VertSize = 6 Pointer.Visible = True XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000

BottomAxis.LabelsFont.Charset
= DEFAULT_CHARSET
BottomAxis.LabelsFont.Color =
clBlack
BottomAxis.LabelsFont.Height
= -11
BottomAxis.LabelsFont.Name =
'Courier New'
BottomAxis.LabelsFont.Style =
[fsBold]
BottomAxis.TickInnerLength = 6
BottomAxis.TicksInner.Width =
2
BottomAxis.Title.Caption =
'Measure Station'
BottomAxis.Title.Font.Charset
= DEFAULT CHARSET
BottomAxis.Title.Font.Color =
clBlue
BottomAxis.Title.Font.Height =
-13
BottomAxis.Title.Font.Name =
'Arial'
BottomAxis.Title.Font.Style =
[fsBold]
LeftAxis.Grid.Color = clBlue
LeftAxis.MinorTickCount = 2
LeftAxis.MinorTickLength = 4
LeftAxis.MinorTickLength = 4 LeftAxis.TickInnerLength = 9
LeftAxis.MinorTickLength = 4 LeftAxis.TickInnerLength = 9 LeftAxis.TickLength = 7
LeftAxis.MinorTickLength = 4 LeftAxis.TickInnerLength = 9 LeftAxis.TickLength = 7 LeftAxis.Ticks.Width = 2
LeftAxis.MinorTickLength = 4 LeftAxis.TickInnerLength = 9 LeftAxis.TickLength = 7 LeftAxis.Ticks.Width = 2 LeftAxis.Title.Caption = 'Liters'
LeftAxis.MinorTickLength = 4 LeftAxis.TickInnerLength = 9 LeftAxis.TickLength = 7 LeftAxis.Ticks.Width = 2 LeftAxis.Title.Caption = 'Liters' LeftAxis.Title.Font.Charset =
LeftAxis.MinorTickLength = 4 LeftAxis.TickInnerLength = 9 LeftAxis.TickLength = 7 LeftAxis.Ticks.Width = 2 LeftAxis.Title.Caption = 'Liters' LeftAxis.Title.Font.Charset = DEFAULT_CHARSET
LeftAxis.MinorTickLength = 4 LeftAxis.TickInnerLength = 9 LeftAxis.TickLength = 7 LeftAxis.Ticks.Width = 2 LeftAxis.Title.Caption = 'Liters' LeftAxis.Title.Font.Charset = DEFAULT_CHARSET LeftAxis.Title.Font.Color =
LeftAxis.MinorTickLength = 4 LeftAxis.TickInnerLength = 9 LeftAxis.TickLength = 7 LeftAxis.Ticks.Width = 2 LeftAxis.Title.Caption = 'Liters' LeftAxis.Title.Font.Charset = DEFAULT_CHARSET LeftAxis.Title.Font.Color = clNavy
LeftAxis.MinorTickLength = 4 LeftAxis.TickInnerLength = 9 LeftAxis.TickLength = 7 LeftAxis.Ticks.Width = 2 LeftAxis.Title.Caption = 'Liters' LeftAxis.Title.Font.Charset = DEFAULT_CHARSET LeftAxis.Title.Font.Color = clNavy LeftAxis.Title.Font.Height = -16
LeftAxis.MinorTickLength = 4 LeftAxis.TickInnerLength = 9 LeftAxis.TickLength = 7 LeftAxis.Ticks.Width = 2 LeftAxis.Title.Caption = 'Liters' LeftAxis.Title.Font.Charset = DEFAULT_CHARSET LeftAxis.Title.Font.Color = clNavy LeftAxis.Title.Font.Height = -16 LeftAxis.Title.Font.Name =
LeftAxis.MinorTickLength = 4 LeftAxis.TickInnerLength = 9 LeftAxis.TickLength = 7 LeftAxis.Ticks.Width = 2 LeftAxis.Title.Caption = 'Liters' LeftAxis.Title.Font.Charset = DEFAULT_CHARSET LeftAxis.Title.Font.Color = clNavy LeftAxis.Title.Font.Height = -16 LeftAxis.Title.Font.Name = 'Arial' LeftAxis.Title.Font.Style = [fsBold, fsItalic]
LeftAxis.MinorTickLength = 4 LeftAxis.TickInnerLength = 9 LeftAxis.TickLength = 7 LeftAxis.Ticks.Width = 2 LeftAxis.Title.Caption = 'Liters' LeftAxis.Title.Font.Charset = DEFAULT_CHARSET LeftAxis.Title.Font.Color = clNavy LeftAxis.Title.Font.Height = -16 LeftAxis.Title.Font.Name = 'Arial' LeftAxis.Title.Font.Style = [fsBold, fsItalic] Legend.Color = clBlue
LeftAxis.MinorTickLength = 4 LeftAxis.TickInnerLength = 9 LeftAxis.TickLength = 7 LeftAxis.Ticks.Width = 2 LeftAxis.Title.Caption = 'Liters' LeftAxis.Title.Font.Charset = DEFAULT_CHARSET LeftAxis.Title.Font.Color = clNavy LeftAxis.Title.Font.Height = -16 LeftAxis.Title.Font.Name = 'Arial' LeftAxis.Title.Font.Style = [fsBold, fsItalic] Legend.Color = clBlue Legend.ColorWidth = 27
LeftAxis.MinorTickLength = 4 LeftAxis.TickInnerLength = 9 LeftAxis.TickLength = 7 LeftAxis.Ticks.Width = 2 LeftAxis.Title.Caption = 'Liters' LeftAxis.Title.Font.Charset = DEFAULT_CHARSET LeftAxis.Title.Font.Color = clNavy LeftAxis.Title.Font.Height = -16 LeftAxis.Title.Font.Name = 'Arial' LeftAxis.Title.Font.Style = [fsBold, fsItalic] Legend.Color = clBlue Legend.ColorWidth = 27 Legend.Font.Charset =
LeftAxis.MinorTickLength = 4 LeftAxis.TickInnerLength = 9 LeftAxis.TickLength = 7 LeftAxis.Ticks.Width = 2 LeftAxis.Title.Caption = 'Liters' LeftAxis.Title.Font.Charset = DEFAULT_CHARSET LeftAxis.Title.Font.Color = clNavy LeftAxis.Title.Font.Height = -16 LeftAxis.Title.Font.Name = 'Arial' LeftAxis.Title.Font.Style = [fsBold, fsItalic] Legend.Color = clBlue Legend.ColorWidth = 27 Legend.Font.Charset = DEFAULT_CHARSET
LeftAxis.MinorTickLength = 4 LeftAxis.TickInnerLength = 9 LeftAxis.TickLength = 7 LeftAxis.Ticks.Width = 2 LeftAxis.Title.Caption = 'Liters' LeftAxis.Title.Font.Charset = DEFAULT_CHARSET LeftAxis.Title.Font.Color = clNavy LeftAxis.Title.Font.Height = -16 LeftAxis.Title.Font.Name = 'Arial' LeftAxis.Title.Font.Style = [fsBold, fsItalic] Legend.Color = clBlue Legend.ColorWidth = 27 Legend.Font.Charset = DEFAULT_CHARSET Legend.Font.Color = clWhite
LeftAxis.MinorTickLength = 4 LeftAxis.TickInnerLength = 9 LeftAxis.TickLength = 7 LeftAxis.Ticks.Width = 2 LeftAxis.Title.Caption = 'Liters' LeftAxis.Title.Font.Charset = DEFAULT_CHARSET LeftAxis.Title.Font.Color = clNavy LeftAxis.Title.Font.Height = -16 LeftAxis.Title.Font.Name = 'Arial' LeftAxis.Title.Font.Style = [fsBold, fsItalic] Legend.Color = clBlue Legend.ColorWidth = 27 Legend.Font.Charset = DEFAULT_CHARSET

1
end
end
<> object TPage
Left = 0
Top = 0
Caption = 'Bar'
<> object Chart2: TChart
Left = 0
Top = 0
Width = 453
Height = 367
BackWall.Brush.Color =
clWhite
BackWall.Brush.Style =
bsClear
BackWall.Color = 8454143
BackWall.Pen.Color = clRed
BottomWall.Size = 4
LeftWall.Color = 16777088
LeftWall.Size = 4
Title.Alignment =
taRightJustify
Title.Font.Charset =
DEFAULT_CHARSET
Title.Font.Color = clBlue
Title.Font.Height = -16
Title.Font.Name = 'Arial'
Title.Font.Style = []
Title.Text.Strings = (
'American Charting Suppliers,
Inc.'
'1997 Production Volume')
BackColor = 8454143
Frame.Color = clRed
LeftAxis.Grid.Visible = False
LeftAxis.Title.Caption =
'Millions of \$'
LeftAxis.Title.Font.Charset =
DEFAULT CHARSET
LeftAxis.Title.Font.Color =
clBlack
LeftAxis.Title.Font.Height = -15 LeftAxis.Title.Font.Name =
'Arial'
LeftAxis.Title.Font.Style = []
Legend.Alignment = laTop
Legend.ColorWidth = 22
Legend.DividingLines.Color =
clSilver

YValues.Order = loNone
end
<> object Series3:
TLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clYellow
Title = 'Asia'
LinePen.Visible = False
Pointer.Brush.Color = clBlue
Pointer.HorizSize = 6
Pointer InflateMargins = False
Pointer.Style =
psDownTriangle
Pointer.VertSize = 6
Pointer. VertSize = 0
Stairs = True
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.00000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
end
<> object Series4:
TLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clBlue
Title = 'Africa'
Pointer.Brush.Color = 8454143
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = True
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.00000000000000000
YValues.Order = loNone
end

XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Bar'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
end
<> object Series6:
TBarSeries
Marks.ArrowLength = 20
Marks.BackColor = 65408
Marks.Visible = True
SeriesColor = clBlue
Title = 'Paris'
MultiBar = mbNone
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Bar'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
end
end
end
<> object TPage
Left = 0
Top = 0
Caption = 'Horizontal Bar'
<> object Chart3: TChart
Left = 0
Top = 0
Width = 453
Height = 367
BackWall.Brush.Color =
clWhite
BackWall.Brush.Style =
bsClear
BackWall.Color = 12615935
BottomWall.Size = 8
Foot.Font.Charset =
DEFAULT_CHARSET

Legend.DividingLines.Visible = True
Legend.ShadowColor = clGray
Legend.ShadowSize = 5
Align = alClient
TabOrder = 0
<> object ComboBox1:
TComboBox
Left = 12
Top = 8
<b>Width = 109</b>
Height = 21
Style = csDropDownList
ItemHeight = 13
TabOrder = 0
OnChange =
ComboBox1Change
Items.Strings = ( 'None'
'Side to Side'
'Stacked'
'Stacked 100%')
end
<> object Series5:
TBarSeries
Marks.ArrowLength = 20
Marks.Visible = False
SeriesColor = clNavy
Title = 'San Francisco'
MultiBar = mbNone
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000000000000000000000
XValues.Order = loAscending
- B
YValues.DateTime = False
YValues.Name = 'Bar'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
end
<> object Series7:
TBarSeries
Marks.ArrowLength = 20
Marks.BackColor = 16777088
Marks.Visible = False
SeriesColor = clYellow
Title = 'Hong Kong'
MultiBar = mbNone
Munipar – monone

THorizBarSeries
Marks.ArrowLength = 20
Marks.Visible = False
SeriesColor = clRed
Title = 'IP'
VertAxis = aRightAxis
MultiBar = mbStacked
XValues.DateTime = False
XValues.Name = 'Bar'
XValues.Multiplier =
1.00000000000000000
XValues.Order = loNone
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.00000000000000000
YValues.Order = loNone
end
<> object Series10:
THorizBarSeries
Marks.ArrowLength = 20
Marks.Visible = False
SeriesColor = clYellow
Title = 'Netware'
VertAxis = aRightAxis
MultiBar = mbStacked
XValues.DateTime = False
XValues.Name = 'Bar'
XValues.Multiplier =
1.00000000000000000
XValues.Order = loNone
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.00000000000000000
YValues.Order = loNone
end
<> object Series9:
THorizBarSeries
Marks.ArrowLength = 20
Marks.BackColor = clBlue
Marks.Font.Charset =
DEFAULT_CHARSET
Marks.Font.Color = clAqua
Marks.Font.Height = -11
Marks.Font.Name = 'Arial'
Marks.Font.Style = []
Marks.Frame.Color = clWhite
Marks.Visible = True
141a1 K3. 4 ISIDIC — 11 UC

Foot.Font.Color = clRed
Foot.Font.Height = -19
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsItalic]
Foot.Text.Strings = (
'Ethernet Protocol Statistics')
LeftWall.Size = 8
MarginLeft = 4
MarginRight = 4
MarginTop = 9
Title.Text.Strings = (
")
Title.Visible = False
BackColor = 12615935
Chart3DPercent = 40
Legend.Alignment = laTop
Legend.TopPos = 19
RightAxis.Title.Caption = 'Hub'
TopAxis.Title.Caption =
'Packets (in thousands)'
TopAxis.Title.Font.Charset =
DEFAULT_CHARSET
TopAxis.Title.Font.Color =
clNavy
TopAxis.Title.Font.Height = -15
TopAxis.Title.Font.Name =
'Arial'
TopAxis.Title.Font.Style =
[fsBold, fsItalic]
Align = alClient
TabOrder = 0
<> object ComboBox2:
TComboBox
Left = 12
Top = 8
Width = 109
Height = 21
Style = csDropDownList
ItemHeight = 13
TabOrder = 0
OnChange =
ComboBox1Change
Items.Strings = (
'None'
'Side to Side'
'Stacked'
'Stacked 100%')
end
<> object Series8:

BottomAxis.Grid.Style =
psSolid Chart3DPercent = 30
LeftAxis.Grid.Style = psSolid
Legend.Color = 8454143
Legend.Font.Charset =
DEFAULT_CHARSET
Legend.Font.Color = clBlack
Legend.Font.Height = -12
Legend.Font.Name = 'Arial'
Legend.Font.Style = [fsBold]
Legend.Frame.Width = 2
RightAxis.Grid.Color =
clYellow
RightAxis.Grid.Style = psSolid
RightAxis.LabelsFont.Charset =
DEFAULT_CHARSET
RightAxis.LabelsFont.Color =
clBlack
RightAxis.LabelsFont.Height =
-11
RightAxis.LabelsFont.Name =
'Arial'
RightAxis.LabelsFont.Style =
[fsBold]
RightAxis.Title.Caption =
'Price'
TopAxis.Grid.Style = psSolid
TopAxis.Grid.Visible = False
TopAxis.LabelsFont.Charset =
DEFAULT_CHARSET
TopAxis.LabelsFont.Color =
clPurple
TopAxis.LabelsFont.Height = -
12
TopAxis.LabelsFont.Name =
'Arial'
TopAxis.LabelsFont.Style = []
Align = alClient
TabOrder = 0
<> object CheckBox4:
TCheckBox
Left = 8
Top = 8
Width = 57
Height = 17
Caption = 'Steps'
TabOrder = 0
OnClick = CheckBox4Click

SeriesColor = 8454016
Title = 'ARP'
VertAxis = aRightAxis
MultiBar = mbStacked
XValues.DateTime = False
XValues.Name = 'Bar'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loNone
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.00000000000000000
YValues.Order = loNone
end
end
end
<> object TPage
Left = 0
Top = 0
Caption = 'Area'
<> object Chart4: TChart
Left = 0
Top = 0
Width = 453
Height = 374
BackWall.Brush.Color =
clWhite
BackWall.Brush.Style =
bsClear
BottomWall.Color = clNavy
V
BottomWall.Size = 8
Foot.Text.Strings = (
1 1 6 1)
'number of users')
LeftWall.Color = clWhite
LeftWall.Size = 8
MarginBottom = 6
MarginLeft = 6
Title.Font.Charset =
DEFAULT_CHARSET
Title.Font.Color = clBlue
Title.Font.Height = -13
Title.Font.Name = 'Arial'
Title.Font.Style = [fsBold]
Title.Text.Strings = (
'Database licences policy'
'Cost per user')

NAVI O I IN
YValues.Order = loNone
end
end
end
<> object TPage
Left = 0
Top = 0
Caption = 'Point XY'
<> object Chart5: TChart
Left = 0
Top = 0
Width = 453
Height = 374
BackWall.Brush.Color =
clWhite
BackWall.Brush.Style =
bsClear
BackWall.Color = clGray
BottomWall.Color = clSilver
BottomWall.Size = 6
Gradient.Direction =
gdRightLeft
Gradient.Visible = True
LeftWall.Color = clSilver
LeftWall.Size = 6
Title.Font.Charset =
DEFAULT CHARSET
Title.Font.Color = clTeal
Title.Font.Height = -16
Title.Font.Name = 'Arial'
Title.Font.Style = [fsBold]
Title.Text.Strings = (
'Fuel Consumption')
BackColor = clGray
BottomAxis.Grid.Visible =
False
LeftAxis.Grid.Visible = False
LeftAxis.Clu. visible Faise  LeftAxis.LabelsFont.Charset =
DEFAULT CHARSET
LeftAxis.LabelsFont.Color =
clOlive
LeftAxis.LabelsFont.Height = -
11
LeftAxis.LabelsFont.Name =
'Arial'
LeftAxis.LabelsFont.Style =
[fsBold]
Legend.Alignment = laTop
Legend.Color = clBlack

end <	
TAreaSeries  HorizAxis = aTopAxis  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clAqua  Title = 'SQL-Provider 1.2'  VertAxis = aRightAxis  AreaColor = clAqua  AreaLinesPen.Color = clGray  DrawArea = True  Pointer.InflateMargins = True  Pointer.Style = psRectangle  Pointer.Visible = False  XValues.DateTime = False  XValues.Multiplier = 1.000000000000000000000000000000000000	
HorizAxis = aTopAxis Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clAqua Title = 'SQL-Provider 1.2' VertAxis = aRightAxis AreaColor = clAqua AreaLinesPen.Color = clGray DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Multiplier = 1.000000000000000000000000000000000000	•
Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clAqua Title = 'SQL-Provider 1.2' VertAxis = aRightAxis AreaColor = clAqua AreaLinesPen.Color = clGray DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	
Marks.Visible = False SeriesColor = clAqua Title = 'SQL-Provider 1.2' VertAxis = aRightAxis AreaColor = clAqua AreaLinesPen.Color = clGray DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000 XValues.Order = loAscending YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000  YValues.Order = loNone end <> object Series12: TAreaSeries HorizAxis = aTopAxis Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clTeal Title = 'Super-OLAP 2.0' VertAxis = aRightAxis AreaColor = clTeal DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	
SeriesColor = clAqua Title = 'SQL-Provider 1.2' VertAxis = aRightAxis AreaColor = clAqua AreaLinesPen.Color = clGray DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Multiplier = 1.000000000000000000000000000000000000	
Title = 'SQL-Provider 1.2'  VertAxis = aRightAxis  AreaColor = clAqua  AreaLinesPen.Color = clGray  DrawArea = True  Pointer.InflateMargins = True  Pointer.Style = psRectangle  Pointer.Visible = False  XValues.DateTime = False  XValues.Multiplier =  1.000000000000000000  XValues.Order = loAscending  YValues.Name = 'Y'  YValues.Multiplier =  1.000000000000000000  YValues.Order = loNone  end  <> object Series12:  TAreaSeries  HorizAxis = aTopAxis  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clTeal  Title = 'Super-OLAP 2.0'  VertAxis = aRightAxis  AreaColor = clTeal  DrawArea = True  Pointer.InflateMargins = True  Pointer.Style = psRectangle  Pointer.Visible = False  XValues.DateTime = False  XValues.Name = 'X'  XValues.Multiplier =  1.000000000000000000  XValues.Order = loAscending  YValues.Multiplier =  1.00000000000000000000000000000000000	Marks.Visible = False
VertAxis = aRightAxis AreaColor = clAqua AreaLinesPen.Color = clGray DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Multiplier = 1.000000000000000000000000000000000000	SeriesColor = clAqua
AreaColor = clAqua AreaLinesPen.Color = clGray DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	Title = 'SQL-Provider 1.2'
AreaLinesPen.Color = clGray DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.0000000000000000000 XValues.DateTime = False YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000 YValues.Order = loNone end <> object Series12: TAreaSeries HorizAxis = aTopAxis Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clTeal Title = 'Super-OLAP 2.0' VertAxis = aRightAxis AreaColor = clTeal DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000 XValues.Order = loAscending YValues.Name = 'Y' YValues.Multiplier = False YValues.Name = 'Y' YValues.Multiplier =	VertAxis = aRightAxis
DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	AreaColor = clAqua
Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	AreaLinesPen.Color = clGray
Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	DrawArea = True
Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.0000000000000000000 XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000 YValues.Order = loNone end <> object Series12: TAreaSeries HorizAxis = aTopAxis Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clTeal Title = 'Super-OLAP 2.0' VertAxis = aRightAxis AreaColor = clTeal DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.00000000000000000 XValues.Order = loAscending YValues.Name = 'Y' YValues.Multiplier =	Pointer.InflateMargins = True
XValues.Name = 'X' XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	Pointer.Style = psRectangle
XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	Pointer.Visible = False
XValues.Multiplier = 1.000000000000000000000000000000000000	XValues.DateTime = False
1.000000000000000000000000000000000000	XValues.Name = 'X'
XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.0000000000000000000 YValues.Order = loNone end <> object Series12: TAreaSeries HorizAxis = aTopAxis Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clTeal Title = 'Super-OLAP 2.0' VertAxis = aRightAxis AreaColor = clTeal DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000 XValues.Order = loAscending YValues.Name = 'Y' YValues.Multiplier =	XValues.Multiplier =
YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000000000000000000000	1.000000000000000000
YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000000000000000000000	XValues.Order = loAscending
YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000000000000000000000	Ü
YValues.Multiplier = 1.000000000000000000000000000000000000	
1.000000000000000000  YValues.Order = loNone end  <> object Series12: TAreaSeries  HorizAxis = aTopAxis  Marks.ArrowLength = 8  Marks.Visible = False SeriesColor = clTeal Title = 'Super-OLAP 2.0'  VertAxis = aRightAxis  AreaColor = clTeal  DrawArea = True  Pointer.InflateMargins = True  Pointer.Style = psRectangle  Pointer.Visible = False  XValues.DateTime = False  XValues.Name = 'X'  XValues.Multiplier =  1.000000000000000000  XValues.Order = loAscending  YValues.Name = 'Y'  YValues.Name = 'Y'  YValues.Multiplier =	
YValues.Order = loNone end <> object Series12: TAreaSeries  HorizAxis = aTopAxis  Marks.ArrowLength = 8  Marks.Visible = False SeriesColor = clTeal  Title = 'Super-OLAP 2.0'  VertAxis = aRightAxis  AreaColor = clTeal  DrawArea = True  Pointer.InflateMargins = True  Pointer.Style = psRectangle  Pointer.Visible = False  XValues.DateTime = False  XValues.Name = 'X'  XValues.Multiplier =  1.00000000000000000000000000000000000	
end <> object Series12: TAreaSeries  HorizAxis = aTopAxis  Marks.ArrowLength = 8  Marks.Visible = False SeriesColor = clTeal Title = 'Super-OLAP 2.0' VertAxis = aRightAxis AreaColor = clTeal DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.00000000000000000 XValues.Order = loAscending YValues.Name = 'Y' YValues.Name = 'Y' YValues.Multiplier =	YValues.Order = loNone
<> object Series12: TAreaSeries  HorizAxis = aTopAxis  Marks.ArrowLength = 8  Marks.Visible = False SeriesColor = clTeal Title = 'Super-OLAP 2.0'  VertAxis = aRightAxis AreaColor = clTeal DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	
TAreaSeries  HorizAxis = aTopAxis  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clTeal  Title = 'Super-OLAP 2.0'  VertAxis = aRightAxis  AreaColor = clTeal  DrawArea = True  Pointer.InflateMargins = True  Pointer.Style = psRectangle  Pointer.Visible = False  XValues.DateTime = False  XValues.Name = 'X'  XValues.Multiplier =  1.00000000000000000  XValues.Order = loAscending  YValues.Name = 'Y'  YValues.Multiplier =	
HorizAxis = aTopAxis Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clTeal Title = 'Super-OLAP 2.0' VertAxis = aRightAxis AreaColor = clTeal DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.00000000000000000 XValues.Order = loAscending YValues.Name = 'Y' YValues.Multiplier =	, i
Marks.ArrowLength = 8  Marks.Visible = False SeriesColor = clTeal Title = 'Super-OLAP 2.0' VertAxis = aRightAxis AreaColor = clTeal DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.0000000000000000 XValues.Order = loAscending YValues.Name = 'Y' YValues.Multiplier =	
Marks.Visible = False SeriesColor = clTeal Title = 'Super-OLAP 2.0' VertAxis = aRightAxis AreaColor = clTeal DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.00000000000000000 XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier =	
SeriesColor = clTeal Title = 'Super-OLAP 2.0' VertAxis = aRightAxis AreaColor = clTeal DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.0000000000000000 XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Name = 'Y' YValues.Multiplier =	č
Title = 'Super-OLAP 2.0' VertAxis = aRightAxis AreaColor = clTeal DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.0000000000000000 XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier =	
VertAxis = aRightAxis AreaColor = clTeal DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.00000000000000000 XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier =	
AreaColor = clTeal DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.0000000000000000 XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier =	
DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.0000000000000000 XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier =	<u> </u>
Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.00000000000000000 XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier =	
Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.00000000000000000 XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier =	
Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.00000000000000000  XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier =	
XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000  XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier =	
XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000  XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier =	
XValues.Multiplier = 1.000000000000000000  XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier =	
1.000000000000000000000000000000000000	
XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier =	
YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier =	
YValues.Name = 'Y' YValues.Multiplier =	
YValues.Multiplier =	
<u>*</u>	
1.000000000000000000	
	1.0000000000000000000

YValues.Name = 'Y'
YValues.Multiplier =
1.00000000000000000
YValues.Order = loNone
end
<> object Series14:
TPointSeries
Marks.ArrowLength = 0
Marks.Visible = False
SeriesColor = clGreen
Title = 'Bikes'
Pointer.HorizSize = 6
Pointer.InflateMargins = True
Pointer.Style = psCircle
Pointer.VertSize = 6
Pointer.Visible = True
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.00000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.00000000000000000
YValues.Order = loNone
end
<> object Series15:
<b>TPointSeries</b>
Marks.ArrowLength = 0
Marks.Visible = False
SeriesColor = clYellow
Title = 'Trucks'
Pointer.HorizSize = 6
Pointer.InflateMargins = True
Pointer.Style = psTriangle
Pointer.VertSize = 6
Pointer.Visible = True
XValues.DateTime = False
XValues.Name = 'X'
XValues.Nultiplier =
1.000000000000000000000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000000000000000000000
YValues.Order = loNone
end
and

Legend.ColorWidth = 32
Legend.Font.Charset =
DEFAULT_CHARSET
Legend.Font.Color = clAqua
<b>Legend.Font.Height = -12</b>
Legend.Font.Name = 'Arial'
Legend.Font.Style = []
Legend.Frame.Color = clRed
Legend.Frame.Width = 2
Legend.ShadowColor = clWhite
Legend.ShadowSize = 1
RightAxis.Grid.Style = psSolid
RightAxis.LabelsFont.Charset =
DEFAULT CHARSET
RightAxis.LabelsFont.Color =
clNavy
RightAxis.LabelsFont.Height =
-12
RightAxis.LabelsFont.Name =
'Arial'
RightAxis.LabelsFont.Style =
[fsItalic]
TopAxis.Grid.Style = psSolid
TopAxis.LabelsFont.Charset =
DEFAULT CHARSET
TopAxis.LabelsFont.Color =
clBlue
TopAxis.LabelsFont.Height = -
12
TopAxis.LabelsFont.Name =
'Arial'
TopAxis.LabelsFont.Style = []
Align = alClient
TabOrder = 0
<> object Series13:
TPointSeries
Marks.ArrowLength = 0
Marks.Visible = False
SeriesColor = clRed
Title = 'Cars'
Pointer InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = True
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False

HorizAxis = aTopAxis
Marks.ArrowLength = 0
Marks.Visible = False
SeriesColor = clGray
Title = 'Rockets'
VertAxis = aRightAxis
Pointer.InflateMargins = True
Pointer.Style = psDiamond
Pointer.VertSize = 9
Pointer. Visible = True
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.00000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
end
end
end
<> object TPage
Left = 0
Top = 0
Caption = 'Pie'
<> object Chart6: TChart
Left = 0
Top = 0
Width = 453
Height = 374
AllowPanning = pmNone
AllowZoom = False
BackWall.Brush.Color =
clWhite
BackWall.Brush.Style =
bsClear
BackWall.Pen.Visible = False
Foot.Text.Strings = (
'The Furniture Specialists
Intl.')
Gradient.Direction =
gdRightLeft
Gradient.Visible = True
Title.Font.Charset =
DEFAULT CHARSET
Title.Font.Color = clBlue
Title.Font.Height = -16

<> object Series16:
TPointSeries The state of the s
HorizAxis = aTopAxis
Marks.ArrowLength = 0
Marks.Visible = False
SeriesColor = clBlue
Title = 'Boats'
VertAxis = aRightAxis
Pointer.HorizSize = 8
Pointer.InflateMargins = True
Pointer.Style =
psDownTriangle
Pointer.VertSize = 8
Pointer.Visible = True
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
end
<> object Series17:
TPointSeries
Marks.ArrowLength = 0
Marks.Visible = False
SeriesColor = clWhite
Title = 'Planes'
Pointer.HorizSize = 7
Pointer.InflateMargins = True
Pointer.Style = psDiagCross
Pointer.VertSize = 2
Pointer.Visible = True
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.00000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Nultiplier =
1.000000000000000000000000000000000000
YValues.Order = loNone
end
<> object Series18: TPointSeries

Marks.Visible = True
SeriesColor = clRed
OtherSlice.Text = 'Other'
PieValues.DateTime = False
PieValues.Name = 'Pie'
PieValues.Multiplier =
1.000000000000000000
PieValues.Order = loNone
end
end
end
<> object TPage
Left = 0
Top = 0
Caption = 'Fast Line'
<> object Chart7: TChart
Left = 0
Top = 0
Width = 453
Height = 374
BackWall.Brush.Color =
clWhite
BackWall.Brush.Style =
bsClear
Title.AdjustFrame = False
Title.Brush.Color = clBlue
Title.Font.Charset =
DEFAULT CHARSET
Title.Font.Color = clAqua
Title.Font.Height = -19
Title.Font.Name = 'Arial'
Title.Font.Style = []
Title.Frame.Width = 2
Title.Frame.Visible = True
Title.Text.Strings = (
'Traded Volume from 1990 to
1996')
BottomAxis.Title.Caption = 'Trading Sessions'
LeftAxis.Title.Caption =
'Number of Shares'
Legend Color = alCroy
Legend.Color = clGray
Legend.Font.Charset = DEFAULT CHARSET
Legend.Font.Color = clWhite
Legend.Font.Height = -11
Legend.Font.Name = 'Arial'
Legend.Font.Style = []

Title.Font.Style = [] Title.Text.Strings = ( '% Growth Rate ') AxisVisible = False ClipPoints = False Frame.Visible = False Legend.Alignment = laLeft Legend.Color = 8453888 Legend.Frame.Color = clBlue Legend.Frame.Style = psDot Legend.Frame.Width = 2 Legend.ShadowColor = clGray Legend.ShadowSize = 4 Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.Font.Charset = DEFAULT CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	Title.Font.Style =    Title.Text.Strings = (     ''% Growth Rate ') AxisVisible = False ClipPoints = False Frame.Visible = False Legend.Alignment = laLeft Legend.Color = 8453888 Legend.Frame.Color = clBlue Legend.Frame.Style = psDot Legend.Frame.Width = 2 Legend.ShadowColor = clGray Legend.ShadowSize = 4 Legend.TextStyle =   ItsLeftPercent Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite	
Title.Text.Strings = (     '% Growth Rate ')     AxisVisible = False     ClipPoints = False     Frame.Visible = False     Legend.Alignment = laLeft     Legend.Color = 8453888     Legend.Frame.Color = clBlue     Legend.Frame.Style = psDot     Legend.Frame.Width = 2     Legend.ShadowColor = clGray     Legend.ShadowSize = 4     Legend.TextStyle = ltsLeftPercent     Legend.TopPos = 59     View3DOptions.Elevation = 315     View3DOptions.Orthogonal = False     View3DOptions.Perspective = 0     View3DOptions.Rotation = 360     View3DWalls = False     Align = alClient     TabOrder = 0     <> object CheckBox3: TCheckBox     Left = 16     Top = 20     Width = 57     Height = 17     Caption = 'Circled'     Color = clYellow     ParentColor = False     TabOrder = 0     OnClick = CheckBox3Click     end     <> object Series19: TPieSeries     Marks.Arrow.Color = clBlack     Marks.ArrowLength = 8     Marks.BackColor = clBlack     Marks.Font.Charset = DEFAULT CHARSET     Marks.Font.Height = -11	Title.Text.Strings = (  '% Growth Rate ')  AxisVisible = False  ClipPoints = False  Frame.Visible = False  Legend.Alignment = laLeft  Legend.Color = 8453888  Legend.Frame.Color = clBlue  Legend.Frame.Style = psDot  Legend.Frame.Width = 2  Legend.ShadowColor = clGray  Legend.ShadowSize = 4  Legend.TextStyle =   ItsLeftPercent  Legend.TopPos = 59  View3DOptions.Derspective = 0  View3DOptions.Perspective = 0  View3DOptions.Rotation = 360  View3DWalls = False  Align = alClient  TabOrder = 0  <> object CheckBox3:  TCheckBox  Left = 16  Top = 20  Width = 57  Height = 17  Caption = 'Circled'  Color = clYellow  ParentColor = False  TabOrder = 0  OnClick = CheckBox3Click  end  <> object Series19:  TPieSeries  Marks.Arrow.Color = clBlack  Marks.ArrowLength = 8  Marks.BackColor = clBlack  Marks.Font.Charset =  DEFAULT CHARSET  Marks.Font.Height = -11  Marks.Font.Name = 'Arial'	Title.Font.Name = 'Arial'
'% Growth Rate ') AxisVisible = False ClipPoints = False Frame.Visible = False Legend.Alignment = laLeft Legend.Color = 8453888 Legend.Frame.Color = clBlue Legend.Frame.Style = psDot Legend.Frame.Width = 2 Legend.ShadowColor = clGray Legend.ShadowSize = 4 Legend.TextStyle = ltsLeftPercent Legend.TopPos = 59 View3DOptions.Devation = 315 View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.Font.Charset = DEFAULT CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	'% Growth Rate ') AxisVisible = False ClipPoints = False Frame.Visible = False Legend.Alignment = laLeft Legend.Color = 8453888 Legend.Frame.Color = clBlue Legend.Frame.Style = psDot Legend.Frame.Width = 2 Legend.ShadowColor = clGray Legend.ShadowSize = 4 Legend.TextStyle = ltsLeftPercent Legend.TopPos = 59 View3DOptions.Derthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Name = 'Arial'	
AxisVisible = False ClipPoints = False Frame.Visible = False Legend.Alignment = laLeft Legend.Color = 8453888 Legend.Frame.Color = clBlue Legend.Frame.Style = psDot Legend.Frame.Width = 2 Legend.ShadowColor = clGray Legend.ShadowSize = 4 Legend.TextStyle = ltsLeftPercent Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.Pont.Charset = DEFAULT CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	AxisVisible = False ClipPoints = False Frame.Visible = False Legend.Alignment = laLeft Legend.Color = 8453888 Legend.Frame.Color = clBlue Legend.Frame.Style = psDot Legend.Frame.Width = 2 Legend.ShadowColor = clGray Legend.ShadowSize = 4 Legend.TextStyle = ltsLeftPercent Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11 Marks.Font.Name = 'Arial'	
ClipPoints = False Frame.Visible = False Legend.Alignment = laLeft Legend.Color = 8453888 Legend.Frame.Color = clBlue Legend.Frame.Style = psDot Legend.Frame.Width = 2 Legend.ShadowColor = clGray Legend.ShadowSize = 4 Legend.TextStyle = ltsLeftPercent Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Color = clWhite Marks.Font.Color = clWhite Marks.Font.Color = clWhite	ClipPoints = False Frame.Visible = False Legend.Alignment = laLeft Legend.Color = 8453888 Legend.Frame.Color = clBlue Legend.Frame.Style = psDot Legend.Frame.Width = 2 Legend.ShadowColor = clGray Legend.ShadowSize = 4 Legend.TextStyle = ltsLeftPercent Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Height = -11 Marks.Font.Name = 'Arial'	,
Frame.Visible = False Legend.Alignment = laLeft Legend.Color = 8453888 Legend.Frame.Color = clBlue Legend.Frame.Style = psDot Legend.Frame.Width = 2 Legend.ShadowColor = clGray Legend.ShadowSize = 4 Legend.TextStyle = ltsLeftPercent Legend.TopPos = 59 View3DOptions.Drthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	Frame.Visible = False Legend.Alignment = laLeft Legend.Color = 8453888 Legend.Frame.Color = clBlue Legend.Frame.Style = psDot Legend.Frame.Width = 2 Legend.ShadowColor = clGray Legend.ShadowSize = 4 Legend.TextStyle = ltsLeftPercent Legend.TopPos = 59 View3DOptions.Drthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT CHARSET Marks.Font.Name = 'Arial'	
Legend.Color = 8453888  Legend.Frame.Color = clBlue  Legend.Frame.Style = psDot  Legend.Frame.Width = 2  Legend.ShadowColor = clGray  Legend.ShadowSize = 4  Legend.TextStyle = ltsLeftPercent  Legend.TopPos = 59  View3DOptions.Elevation = 315  View3DOptions.Orthogonal = False  View3DOptions.Perspective = 0  View3DOptions.Rotation = 360  View3DWalls = False  Align = alClient  TabOrder = 0  <> object CheckBox3:  TCheckBox  Left = 16  Top = 20  Width = 57  Height = 17  Caption = 'Circled'  Color = clYellow  ParentColor = False  TabOrder = 0  OnClick = CheckBox3Click  end  <> object Series19:  TPieSeries  Marks.Arrow.Color = clBlack  Marks.ArrowLength = 8  Marks.BackColor = clBlack  Marks.Font.Charset =  DEFAULT_CHARSET  Marks.Font.Height = -11	Legend.Alignment = laLeft Legend.Color = 8453888 Legend.Frame.Color = clBlue Legend.Frame.Style = psDot Legend.Frame.Width = 2 Legend.ShadowColor = clGray Legend.ShadowSize = 4 Legend.TextStyle = ltsLeftPercent Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Name = 'Arial'	ClipPoints = False
Legend.Color = 8453888  Legend.Frame.Color = clBlue  Legend.Frame.Style = psDot  Legend.Frame.Width = 2  Legend.ShadowColor = clGray  Legend.ShadowSize = 4  Legend.TextStyle =  ItsLeftPercent  Legend.TopPos = 59  View3DOptions.Elevation = 315  View3DOptions.Orthogonal =  False  View3DOptions.Perspective = 0  View3DOptions.Rotation = 360  View3DWalls = False  Align = alClient  TabOrder = 0  <> object CheckBox3:  TCheckBox  Left = 16  Top = 20  Width = 57  Height = 17  Caption = 'Circled'  Color = clYellow  ParentColor = False  TabOrder = 0  OnClick = CheckBox3Click  end  <> object Series19:  TPieSeries  Marks.Arrow.Color = clBlack  Marks.ArrowLength = 8  Marks.BackColor = clBlack  Marks.Font.Charset =  DEFAULT_CHARSET  Marks.Font.Color = clWhite  Marks.Font.Height = -11	Legend.Color = 8453888  Legend.Frame.Color = clBlue  Legend.Frame.Style = psDot  Legend.Frame.Width = 2  Legend.ShadowColor = clGray  Legend.ShadowSize = 4  Legend.TextStyle =  ItsLeftPercent  Legend.TopPos = 59  View3DOptions.Elevation = 315  View3DOptions.Orthogonal =  False  View3DOptions.Perspective = 0  View3DOptions.Rotation = 360  View3DWalls = False  Align = alClient  TabOrder = 0  <> object CheckBox3:  TCheckBox  Left = 16  Top = 20  Width = 57  Height = 17  Caption = 'Circled'  Color = clYellow  ParentColor = False  TabOrder = 0  OnClick = CheckBox3Click  end  <> object Series19:  TPieSeries  Marks.Arrow.Color = clBlack  Marks.ArrowLength = 8  Marks.BackColor = clBlack  Marks.Font.Charset =  DEFAULT_CHARSET  Marks.Font.Name = 'Arial'	Frame.Visible = False
Legend.Frame.Color = clBlue Legend.Frame.Style = psDot Legend.Frame.Width = 2 Legend.ShadowColor = clGray Legend.ShadowSize = 4 Legend.TextStyle = ltsLeftPercent Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	Legend.Frame.Color = clBlue Legend.Frame.Style = psDot Legend.Frame.Width = 2 Legend.ShadowColor = clGray Legend.ShadowSize = 4 Legend.TextStyle = ltsLeftPercent Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Name = 'Arial'	
Legend.Frame.Color = clBlue Legend.Frame.Style = psDot Legend.Frame.Width = 2 Legend.ShadowColor = clGray Legend.ShadowSize = 4 Legend.TextStyle = ltsLeftPercent Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	Legend.Frame.Color = clBlue Legend.Frame.Style = psDot Legend.Frame.Width = 2 Legend.ShadowColor = clGray Legend.ShadowSize = 4 Legend.TextStyle = ltsLeftPercent Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Name = 'Arial'	<b>Legend.Color = 8453888</b>
Legend.Frame.Width = 2 Legend.ShadowColor = clGray Legend.ShadowSize = 4 Legend.TextStyle = ltsLeftPercent Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	Legend.Frame.Width = 2 Legend.ShadowColor = clGray Legend.ShadowSize = 4 Legend.TextStyle = ItsLeftPercent Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11 Marks.Font.Name = 'Arial'	
Legend.Frame.Width = 2 Legend.ShadowColor = clGray Legend.ShadowSize = 4 Legend.TextStyle = ltsLeftPercent Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	Legend.Frame.Width = 2 Legend.ShadowColor = clGray Legend.ShadowSize = 4 Legend.TextStyle = ItsLeftPercent Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11 Marks.Font.Name = 'Arial'	Legend.Frame.Style = psDot
Legend.ShadowSize = 4 Legend.TextStyle =  ItsLeftPercent  Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal =  False  View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False  Align = alClient  TabOrder = 0 <> object CheckBox3:  TCheckBox  Left = 16  Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False  TabOrder = 0 OnClick = CheckBox3Click end <> object Series19:  TPieSeries  Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset =  DEFAULT_CHARSET  Marks.Font.Color = clWhite Marks.Font.Height = -11	Legend.ShadowSize = 4 Legend.TextStyle = ItsLeftPercent Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Name = 'Arial'	
Legend.ShadowSize = 4 Legend.TextStyle =  ItsLeftPercent  Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal =  False  View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False  Align = alClient  TabOrder = 0 <> object CheckBox3:  TCheckBox  Left = 16  Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False  TabOrder = 0 OnClick = CheckBox3Click end <> object Series19:  TPieSeries  Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset =  DEFAULT_CHARSET  Marks.Font.Color = clWhite Marks.Font.Height = -11	Legend.ShadowSize = 4 Legend.TextStyle = ItsLeftPercent Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Name = 'Arial'	Legend.ShadowColor = clGray
Legend.TextStyle = ltsLeftPercent Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT CHARSET Marks.Font.Height = -11	Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Name = 'Arial'	
ItsLeftPercent Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	ItsLeftPercent  Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False  View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries  Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT CHARSET Marks.Font.Color = clWhite Marks.Font.Name = 'Arial'	
Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT CHARSET Marks.Font.Color = clWhite Marks.Font.Color = clWhite Marks.Font.Color = clWhite	Legend.TopPos = 59 View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Name = 'Arial'	
View3DOptions.Drthogonal = False  View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries  Marks.Arrow.Color = clBlack Marks.Font.Charset = DEFAULT CHARSET Marks.Font.Color = clWhite	View3DOptions.Drthogonal = False  View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Name = 'Arial'	
View3DOptions.Orthogonal = False  View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries  Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	View3DOptions.Orthogonal = False  View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Name = 'Arial'	
View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11 Marks.Font.Name = 'Arial'	
View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Name = 'Arial'	
View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	View3DOptions.Rotation = 360 View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11 Marks.Font.Name = 'Arial'	
View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	View3DWalls = False Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Name = 'Arial'	
Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox  Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries  Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	Align = alClient TabOrder = 0 <> object CheckBox3: TCheckBox  Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11 Marks.Font.Name = 'Arial'	
TabOrder = 0  <> object CheckBox3:  TCheckBox  Left = 16  Top = 20  Width = 57  Height = 17  Caption = 'Circled'  Color = clYellow  ParentColor = False  TabOrder = 0  OnClick = CheckBox3Click end  <> object Series19:  TPieSeries  Marks.Arrow.Color = clBlack  Marks.ArrowLength = 8  Marks.Font.Charset =  DEFAULT_CHARSET  Marks.Font.Color = clWhite  Marks.Font.Height = -11	TabOrder = 0  <> object CheckBox3: TCheckBox  Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries  Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Name = 'Arial'	
<> object CheckBox3: TCheckBox  Left = 16  Top = 20  Width = 57  Height = 17  Caption = 'Circled'  Color = clYellow  ParentColor = False  TabOrder = 0  OnClick = CheckBox3Click  end  <> object Series19: TPieSeries  Marks.Arrow.Color = clBlack  Marks.ArrowLength = 8  Marks.Font.Charset =  DEFAULT_CHARSET  Marks.Font.Color = clWhite  Marks.Font.Height = -11	<> object CheckBox3: TCheckBox  Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Name = 'Arial'	
TCheckBox Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	TCheckBox  Left = 16  Top = 20  Width = 57  Height = 17  Caption = 'Circled'  Color = clYellow  ParentColor = False  TabOrder = 0  OnClick = CheckBox3Click end  <> object Series19:  TPieSeries  Marks.Arrow.Color = clBlack  Marks.ArrowLength = 8  Marks.BackColor = clBlack  Marks.Font.Charset =  DEFAULT CHARSET  Marks.Font.Color = clWhite  Marks.Font.Name = 'Arial'	
Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	Left = 16 Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Name = 'Arial'	•
Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	Top = 20 Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11 Marks.Font.Name = 'Arial'	
Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	Width = 57 Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Name = 'Arial'	
Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	Height = 17 Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11 Marks.Font.Name = 'Arial'	
Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	Caption = 'Circled' Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11 Marks.Font.Name = 'Arial'	
Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	Color = clYellow ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11 Marks.Font.Name = 'Arial'	
ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	ParentColor = False TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11 Marks.Font.Name = 'Arial'	
TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	TabOrder = 0 OnClick = CheckBox3Click end <> object Series19: TPieSeries  Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11 Marks.Font.Name = 'Arial'	
OnClick = CheckBox3Click end <> object Series19: TPieSeries  Marks.Arrow.Color = clBlack Marks.ArrowLength = 8  Marks.BackColor = clBlack Marks.Font.Charset =  DEFAULT_CHARSET  Marks.Font.Color = clWhite Marks.Font.Height = -11	OnClick = CheckBox3Click end <> object Series19: TPieSeries  Marks.Arrow.Color = clBlack Marks.ArrowLength = 8  Marks.BackColor = clBlack Marks.Font.Charset =  DEFAULT_CHARSET  Marks.Font.Color = clWhite Marks.Font.Height = -11  Marks.Font.Name = 'Arial'	
end <> object Series19: TPieSeries  Marks.Arrow.Color = clBlack  Marks.ArrowLength = 8  Marks.BackColor = clBlack  Marks.Font.Charset =  DEFAULT_CHARSET  Marks.Font.Color = clWhite  Marks.Font.Height = -11	end <> object Series19: TPieSeries  Marks.Arrow.Color = clBlack  Marks.ArrowLength = 8  Marks.BackColor = clBlack  Marks.Font.Charset =  DEFAULT_CHARSET  Marks.Font.Color = clWhite  Marks.Font.Height = -11  Marks.Font.Name = 'Arial'	
<> object Series19: TPieSeries  Marks.Arrow.Color = clBlack  Marks.ArrowLength = 8  Marks.BackColor = clBlack  Marks.Font.Charset =  DEFAULT_CHARSET  Marks.Font.Color = clWhite  Marks.Font.Height = -11	<pre>&lt;&gt; object Series19: TPieSeries  Marks.Arrow.Color = clBlack Marks.ArrowLength = 8  Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11 Marks.Font.Name = 'Arial'</pre>	
TPieSeries  Marks.Arrow.Color = clBlack  Marks.ArrowLength = 8  Marks.BackColor = clBlack  Marks.Font.Charset =  DEFAULT_CHARSET  Marks.Font.Color = clWhite  Marks.Font.Height = -11	TPieSeries  Marks.Arrow.Color = clBlack  Marks.ArrowLength = 8  Marks.BackColor = clBlack  Marks.Font.Charset =  DEFAULT_CHARSET  Marks.Font.Color = clWhite  Marks.Font.Height = -11  Marks.Font.Name = 'Arial'	
Marks.Arrow.Color = clBlack Marks.ArrowLength = 8 Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	Marks.Arrow.Color = clBlack Marks.ArrowLength = 8  Marks.BackColor = clBlack Marks.Font.Charset =  DEFAULT_CHARSET  Marks.Font.Color = clWhite  Marks.Font.Height = -11  Marks.Font.Name = 'Arial'	· ·
Marks.ArrowLength = 8  Marks.BackColor = clBlack  Marks.Font.Charset =  DEFAULT_CHARSET  Marks.Font.Color = clWhite  Marks.Font.Height = -11	Marks.ArrowLength = 8  Marks.BackColor = clBlack  Marks.Font.Charset =  DEFAULT_CHARSET  Marks.Font.Color = clWhite  Marks.Font.Height = -11  Marks.Font.Name = 'Arial'	
Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	Marks.BackColor = clBlack Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11 Marks.Font.Name = 'Arial'	
Marks.Font.Charset = DEFAULT_CHARSET Marks.Font.Color = clWhite Marks.Font.Height = -11	Marks.Font.Charset = DEFAULT_CHARSET  Marks.Font.Color = clWhite  Marks.Font.Height = -11  Marks.Font.Name = 'Arial'	
DEFAULT_CHARSET  Marks.Font.Color = clWhite  Marks.Font.Height = -11	DEFAULT_CHARSET  Marks.Font.Color = clWhite  Marks.Font.Height = -11  Marks.Font.Name = 'Arial'	
Marks.Font.Color = clWhite Marks.Font.Height = -11	Marks.Font.Color = clWhite Marks.Font.Height = -11 Marks.Font.Name = 'Arial'	
Marks.Font.Height = -11	Marks.Font.Height = -11 Marks.Font.Name = 'Arial'	
	Marks.Font.Name = 'Arial'	
	Marks.Font.Style = []	
Marks.Font.Style = []		Marks.Font.Style = []

# <u>AC.&P</u>

1.000000000000000000000000000000000000
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
end
<> object Series23:
TFastLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clBlue
Title = 'Barcelona'
LinePen.Color = clBlue
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.00000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000000000000000000000
YValues.Order = loNone
end
<> object Series24:
TFastLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clWhite
Title = 'Montreal'
LinePen.Color = clWhite
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.00000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
end
end
end
end <> object TPage

View3D = False
Align = alClient
TabOrder = 0
<> object Series20:
TFastLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clRed
Title = 'New York'
LinePen.Color = clRed
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000000000000000000000
YValues.Order = loNone
end
<> object Series21:
TFastLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clGreen
Title = 'Chicago'
LinePen.Color = clGreen
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.00000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.00000000000000000
YValues.Order = loNone
end
<> object Series22:
TFastLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clYellow
Title = 'Copenhague'
LinePen.Color = clYellow
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =

EndYValues.Order = loNone
StartXValues.DateTime =
False
StartXValues.Name = 'X'
StartXValues.Multiplier =
1.000000000000000000
StartXValues.Order =
loAscending
StartYValues.DateTime =
False
StartYValues.Name = 'Y'
StartYValues.Multiplier =
1.00000000000000000
StartYValues.Order = loNone
end
<> object Series27:
TArrowSeries
Marks.ArrowLength = 0
Marks.Frame.Visible = False
Marks.Transparent = True
Marks.Visible = False
SeriesColor = clGreen
Pointer.InflateMargins = False
Pointer.Style = psRectangle
Pointer.Visible = True
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
EndXValues.DateTime = True
EndXValues.Name = 'EndX'
EndXValues.Multiplier =
1.000000000000000000
EndXValues.Order = loNone
EndYValues.DateTime = False
EndYValues.Name = 'EndY'
EndYValues.Multiplier =
1.00000000000000000
EndYValues.Order = loNone
StartXValues.DateTime =
True
StartXValues.Name = 'X'
StartXValues.Multiplier =

Caption = 'Arrow'
<> object Chart8: TChart
Left = 0
Top = 0
Width = 453
Height = 374
BackWall.Brush.Color =
clWhite
BackWall.Brush.Style =
bsClear
Title.Font.Charset =
DEFAULT CHARSET
Title.Font.Color = clBlue
Title.Font.Height = -16
Title.Font.Name = 'Arial'
Title.Font.Style = []
Title.Text.Strings = (
'Weather in Japan')
Align = alClient
TabOrder = 0
<> object Series26:
TArrowSeries
Marks.ArrowLength = 0
Marks.Frame.Visible = False
Marks.Transparent = True
Marks.Visible = False
SeriesColor = clRed
Pointer.InflateMargins = False
Pointer.Style = psRectangle
Pointer.Visible = True
XValues.DateTime = False
XValues.Name = 'X'
XValues.Nultiplier =
1.000000000000000000000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
EndXValues.DateTime = True
EndXValues.Name = 'EndX'
EndXValues.Multiplier =
1.0000000000000000000
EndXValues.Order = loNone
EndYValues.DateTime = False
EndYValues.Name = 'EndY'
EndYValues.Multiplier =
1.000000000000000000

1.0000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.00000000000000000
YValues.Order = loNone
RadiusValues.DateTime =
False
RadiusValues.Name = 'Radius'
RadiusValues.Multiplier =
1.000000000000000000
RadiusValues.Order = loNone
end
end
end
<> object TPage
Left = 0
Top = 0
Caption = 'Gantt'
<> object Chart10: TChart
Left = 0
Top = 0
Width = 453
Height = 374
BackWall.Brush.Color =
clWhite
BackWall.Brush.Style =
bsClear
Title.Text.Strings = (
'TChart')
Chart3DPercent = 5
Legend.Visible = False
Align = alClient
TabOrder = 0
<> object Series29:
TGanttSeries
ColorEachPoint = True
Marks.ArrowLength = 0
Marks. Visible = False
SeriesColor = clRed
Pointer InflateMargins = True
Pointer.Style = psRectangle Pointer.Visible = True
XValues.DateTime = True
XValues.Name = 'Start'
XValues.Multiplier =
1.000000000000000000000000000000000000
XValues.Order = loAscending

Font.Color = clBlack
Font.Height = -11
Font.Name = 'Arial'
Font.Style = []
$\mathbf{X0} = 20.00000000000000000000000000000000000$
X1 = 100.00000000000000000000000000000000
Y0 = 20.000000000000000000000000000000000
Y1 = 100.00000000000000000000000000000000
XValues.DateTime = False
XValues.Name = 'X'
XValues.Nultiplier =
1.000000000000000000000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.00000000000000000
YValues.Order = loNone
end
<> object Series31:
<b>TChartShape</b>
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clWhite
Brush.Color = clYellow
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'Arial'
Font.Style = []
Style = chasDiamond
X0 = 30.000000000000000000
X1 = 90.000000000000000000
Y0 = 30.5000000000000000000
Y1 = 114.00000000000000000000000000000000000
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.00000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.00000000000000000
YValues.Order = loNone
end
<> object Series32:
TChartShape
1 Chai tshape

YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
StartValues.DateTime = True
StartValues.Name = 'Start'
StartValues.Multiplier =
1.000000000000000000
StartValues.Order =
loAscending
EndValues.DateTime = False
EndValues.Name = 'End'
EndValues.Multiplier =
1.000000000000000000
EndValues.Order = loNone
NextTask.DateTime = False
NextTask.Name = 'NextTask'
NextTask.Multiplier =
1.000000000000000000
NextTask.Order = loNone
end
end
end
<> object TPage
Left = 0
Top = 0
Caption = 'Shapes'
<> object Chart11: TChart
Left = 0
Top = 0
Width = 453
Height = 374
BackWall.Brush.Color =
clWhite
BackWall.Brush.Style =
bsClear
Title.Text.Strings = (
'TChart')
,
Align = alClient
TabOrder = 0
<pre>&lt;&gt; object Series30:</pre>
TChartShape
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clWhite
Brush.Color = clWhite
Font.Charset =
DEFAULT_CHARSET

# <u>AC.&P</u>

D (F ( F)
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Chart1: TChart
Left = 161
Top = 0
Width = 469
Height = 404
AnimatedZoom = True
AnimatedZoomSteps = 16
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Color = 8454016
BackWall.Pen.Color = clNavy
BackWall.Pen.Width = 2
BottomWall.Size = 6
Foot.Frame.Color = clScrollBar
LeftWall.Size = 4
Title.Frame.Color = clScrollBar
Title.Text.Strings = (
'We''re testing that new
Pentium'
'It seems has good health !!!')
BackColor = 8454016
BottomAxis.Grid.Color = 8388863
Frame.Color = clNavy
Frame.Width = 2
LeftAxis.Grid.Color = clSilver
Legend.Font.Charset =
DEFAULT CHARSET
Legend.Font.Color = clBlack
Legend.Font.Height = -19
Legend.Font.Name = 'Arial'
Legend.Font.Style = [fsBold,
fsItalic]
Legend.LegendStyle =
lsLastValues
OnGetLegendText =
Chart1GetLegendText
Align = alClient
TabOrder = 0
<> object LineSeries1:
TLineSeries TLineSeries TLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
1.202 2207 1 22020 I WIDO

Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clWhite
Brush.Color = clAqua
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'Arial'
Font.Style = []
Style = chasTriangle
X0 = 30.000000000000000000000000000000000
X1 = 90.0000000000000000000000000000000000
Y0 = 47.5000000000000000000000000000000000000
Y1 = 102.00000000000000000000000000000000000
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.00000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
end
end
end
end
<> object Timer1: TTimer
Enabled = False
Interval = 300
OnTimer = Timer1Timer
Left = 404
Top = 18
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
21
<> object DigitalForm:
TDigitalForm
Left = 196
Top = 113
Width = 638
Height = 438
Caption = 'TeeChart Digital Series
Example'
Color = clBtnFace

# <u>AC.&P</u>

Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
Stairs = True
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.0000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000000000000000000000
YValues.Order = loNone
Left = 107
Top = 86
end
<> object LineSeries4:
TLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clRed
Title = 'I/O Timing'
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
Stairs = True
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 173
Top = 88
end
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 161
Height = 404
Align = alLeft
TabOrder = 1
10001001

SeriesColor = clBlue
Title = 'Pipeline Signal'
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
Stairs = True
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.00000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000000000000000000000
YValues.Order = loNone Left = 175
Top = 130
end
<> object LineSeries2:
TLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clYellow
Title = 'RAM Cycles'
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
Stairs = True
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.00000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000000000000000000000
YValues.Order = loNone
Left = 105
Top = 132
end
<> object LineSeries3:
TLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clGreen
Title = 'PCI Status'

Kind = bkClose
end
<> object CheckBox4:
TCheckBox
Left = 20
Top = 93 Width = 78
Height = 18
Caption = 'Z Order'
TabOrder = 4
OnClick = CheckBox4Click
end
<> object Memo1: TMemo
Left = 20
Top = 136
Width = 121
Height = 169
Lines.Strings = (
'Line Series can be used '
'as "step" Series by '
'setting '
'the Stairs property.'
"
'In this demo several '
'Line Series are '
'displayed '
'on same Z order.')
TabOrder = 5
end
end
<> object Timer1: TTimer
Enabled = False
Interval = 1
OnTimer = Timer1Timer
Left = 170
Top = 10
end
End
\\\\\\\\
22
<> object DrawForm:
TDrawForm
Left = 182
Top = 109
Width = 599
Height = 398
Caption = 'TeeChart Customized
-
Drawing Example'

<> object CheckBox1:
TCheckBox
Left = 20
Top = 13
Width = 119
Height = 18
Caption = 'Animate !!!'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -19
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 0
OnClick = CheckBox1Click
end
<> object CheckBox2:
TCheckBox
Left = 20
Top = 69
<b>Width</b> = 78
Height = 18
Caption = 'View 3D'
Checked = True
State = cbChecked
TabOrder = 1
OnClick = CheckBox2Click
end
<> object CheckBox3:
TCheckBox
Left = 20
Top = 45
Width = 126
Height = 18
Caption = 'Last Values in
Legend'
Checked = True
State = cbChecked
TabOrder = 2
OnClick = CheckBox3Click
end
<> object BitBtn3: TBitBtn
Left = 32
Top = 351
Width = 89
Height = 33
TabOrder = 3
l.

# <u>AC.&P</u>

XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.0000000000000000000
YValues.Order = loNone
Left = 199
Top = 122
end
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 145
Height = 364
Align = alLeft
TabOrder = 1
<> object BitBtn3: TBitBtn
Left = 24
Top = 312
Width = 89
11 11 11
Height = 33 TabOrder = 0
Kind = bkClose
end
<> object CheckBox1:
TCheckBox
Left = 16
Top = 12
Width = 117
Height = 17
Caption = '&Animate !!!'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -19
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 1
OnClick = CheckBox1Click
end
<> object BitBtn1: TBitBtn
Left = 16

Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Chart1: TChart
Left = 145
Top = 0
Width = 446
Height = 364
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Color = clSilver
BottomWall.Color = 16777088
BottomWall.Size = 4
Foot.Font.Charset =
DEFAULT CHARSET
Foot.Font.Color = clRed
Foot.Font.Height = -12
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsItalic]
Foot.Frame.Color = clScrollBar
LeftWall.Color = 8454016
LeftWall.Size = 4
Title.Frame.Color = clScrollBar
Title.Text.Strings = (
'TChart Component')
BackColor = clSilver
Chart3DPercent = 65
Legend.ColorWidth = 32
Align = alClient
Color = clWhite
TabOrder = 0
<> object LineSeries1:
TLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clBlue
AfterDrawValues =
LineSeries1AfterDrawValues
BeforeDrawValues =
LineSeries1BeforeDrawValues
LinePen.Width = 3
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
i diliter v isidie – Faise

TFastLineForm Left = 192 Top = 108 Width = 575 Height = 415 Caption = 'TeeChart Fast LineSeries Example' Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 0 Width = 414 Height = 381 AnimatedZoom = True AnimatedZoomSteps = 15 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Pen.Color = clBlue BackWall.Pen.Visible = False Foot.Font.Charset = DEFAULT_CHARSET Foot.Font.Color = clRed
Top = 108 Width = 575 Height = 415 Caption = 'TeeChart Fast LineSeries Example' Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 0 Width = 414 Height = 381 AnimatedZoom = True AnimatedZoomSteps = 15 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Pen.Color = clBlue BackWall.Pen.Visible = False Foot.Font.Charset = DEFAULT_CHARSET
Width = 575 Height = 415 Caption = 'TeeChart Fast LineSeries Example' Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 0 Width = 414 Height = 381 AnimatedZoom = True AnimatedZoomSteps = 15 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Pen.Color = clBlue BackWall.Pen.Visible = False Foot.Font.Charset = DEFAULT_CHARSET
Height = 415 Caption = 'TeeChart Fast LineSeries Example' Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 0 Width = 414 Height = 381 AnimatedZoom = True AnimatedZoomSteps = 15 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Pen.Color = clBlue BackWall.Pen.Visible = False Foot.Font.Charset = DEFAULT_CHARSET
Caption = 'TeeChart Fast LineSeries Example'  Color = clBtnFace  ParentFont = True  OldCreateOrder = True  Position = poScreenCenter  OnCreate = FormCreate  PixelsPerInch = 96  TextHeight = 13  <> object Chart1: TChart  Left = 153  Top = 0  Width = 414  Height = 381  AnimatedZoom = True  AnimatedZoomSteps = 15  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  BackWall.Pen.Color = clBlue  BackWall.Pen.Visible = False  Foot.Font.Charset =  DEFAULT_CHARSET
LineSeries Example' Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 0 Width = 414 Height = 381 AnimatedZoom = True AnimatedZoomSteps = 15 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Pen.Color = clBlue BackWall.Pen.Visible = False Foot.Font.Charset = DEFAULT_CHARSET
LineSeries Example' Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 0 Width = 414 Height = 381 AnimatedZoom = True AnimatedZoomSteps = 15 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Pen.Color = clBlue BackWall.Pen.Visible = False Foot.Font.Charset = DEFAULT_CHARSET
Color = clBtnFace ParentFont = True OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 0 Width = 414 Height = 381 AnimatedZoom = True AnimatedZoomSteps = 15 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Pen.Color = clBlue BackWall.Pen.Visible = False Foot.Font.Charset = DEFAULT_CHARSET
ParentFont = True OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 0 Width = 414 Height = 381 AnimatedZoom = True AnimatedZoomSteps = 15 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Pen.Color = clBlue BackWall.Pen.Visible = False Foot.Font.Charset = DEFAULT_CHARSET
OldCreateOrder = True Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 0 Width = 414 Height = 381 AnimatedZoom = True AnimatedZoomSteps = 15 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Pen.Color = clBlue BackWall.Pen.Visible = False Foot.Font.Charset = DEFAULT_CHARSET
Position = poScreenCenter OnCreate = FormCreate PixelsPerInch = 96 TextHeight = 13 <> object Chart1: TChart Left = 153 Top = 0 Width = 414 Height = 381 AnimatedZoom = True AnimatedZoomSteps = 15 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Pen.Color = clBlue BackWall.Pen.Visible = False Foot.Font.Charset = DEFAULT_CHARSET
OnCreate = FormCreate  PixelsPerInch = 96  TextHeight = 13  <> object Chart1: TChart  Left = 153  Top = 0  Width = 414  Height = 381  AnimatedZoom = True  AnimatedZoomSteps = 15  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  BackWall.Pen.Color = clBlue  BackWall.Pen.Visible = False  Foot.Font.Charset =  DEFAULT_CHARSET
PixelsPerInch = 96  TextHeight = 13  <> object Chart1: TChart  Left = 153  Top = 0  Width = 414  Height = 381  AnimatedZoom = True  AnimatedZoomSteps = 15  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  BackWall.Pen.Color = clBlue  BackWall.Pen.Visible = False  Foot.Font.Charset =  DEFAULT_CHARSET
TextHeight = 13  <> object Chart1: TChart  Left = 153  Top = 0  Width = 414  Height = 381  AnimatedZoom = True  AnimatedZoomSteps = 15  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  BackWall.Pen.Color = clBlue  BackWall.Pen.Visible = False  Foot.Font.Charset =  DEFAULT_CHARSET
<> object Chart1: TChart  Left = 153  Top = 0  Width = 414  Height = 381  AnimatedZoom = True  AnimatedZoomSteps = 15  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  BackWall.Pen.Color = clBlue  BackWall.Pen.Visible = False  Foot.Font.Charset =  DEFAULT_CHARSET
Left = 153 Top = 0 Width = 414 Height = 381 AnimatedZoom = True AnimatedZoomSteps = 15 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Pen.Color = clBlue BackWall.Pen.Visible = False Foot.Font.Charset = DEFAULT_CHARSET
Top = 0  Width = 414  Height = 381  AnimatedZoom = True  AnimatedZoomSteps = 15  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  BackWall.Pen.Color = clBlue  BackWall.Pen.Visible = False  Foot.Font.Charset =  DEFAULT_CHARSET
Width = 414 Height = 381 AnimatedZoom = True AnimatedZoomSteps = 15 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Pen.Color = clBlue BackWall.Pen.Visible = False Foot.Font.Charset = DEFAULT_CHARSET
Height = 381  AnimatedZoom = True  AnimatedZoomSteps = 15  BackWall.Brush.Color = clWhite  BackWall.Brush.Style = bsClear  BackWall.Pen.Color = clBlue  BackWall.Pen.Visible = False  Foot.Font.Charset =  DEFAULT_CHARSET
AnimatedZoom = True AnimatedZoomSteps = 15 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Pen.Color = clBlue BackWall.Pen.Visible = False Foot.Font.Charset = DEFAULT_CHARSET
AnimatedZoomSteps = 15 BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Pen.Color = clBlue BackWall.Pen.Visible = False Foot.Font.Charset = DEFAULT_CHARSET
BackWall.Brush.Color = clWhite BackWall.Brush.Style = bsClear BackWall.Pen.Color = clBlue BackWall.Pen.Visible = False Foot.Font.Charset = DEFAULT_CHARSET
BackWall.Brush.Style = bsClear BackWall.Pen.Color = clBlue BackWall.Pen.Visible = False Foot.Font.Charset = DEFAULT_CHARSET
BackWall.Pen.Color = clBlue BackWall.Pen.Visible = False Foot.Font.Charset = DEFAULT_CHARSET
BackWall.Pen.Visible = False Foot.Font.Charset = DEFAULT_CHARSET
Foot.Font.Charset = DEFAULT_CHARSET
DEFAULT_CHARSET
<u> </u>
Foot Font Color = clRed
i juli unicului ciillu
Foot.Font.Height = -12
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsItalic]
Foot.Frame.Color = clScrollBar
Foot.Visible = False
MarginBottom = 1
MarginLeft = 1
MarginRight = 1
MarginTop = 1
Title.Frame.Color = clScrollBar
Title.Text.Strings = (
'TChart Component')
Title.Visible = False
AxisVisible = False
BottomAxis.Grid.Color = 8421440
<b>BottomAxis.LabelsFont.Charset =</b>
DEFAULT_CHARSET
<b>BottomAxis.LabelsFont.Color =</b>
clFuchsia
BottomAxis.LabelsFont.Height = -

_
Top = 40
Width = 89
Height = 33
Caption = '&Print !'
TabOrder = 2
OnClick = BitBtn1Click
Glyph.Data = {
NumGlyphs = 2
end
<> object CheckBox2:
TCheckBox
Left = 16
Top = 84
Width = 57
Height = 17
Caption = '3D'
Checked = True
State = cbChecked
TabOrder = 3
OnClick = CheckBox2Click
end
<> object Memo1: TMemo
Left = 12
Top = 116
Width = 121
Height = 169
Lines.Strings = (
'Custom drawing using '
'the Chart Canvas '
'property is easy !'
11
'This demo shows how '
'custom drawings can be '
'printed too.')
TabOrder = 4
end
end
<> object Timer1: TTimer
Enabled = False
Interval = 1
OnTimer = Timer1Timer
Left = 642
Top = 37
end
End
\\\\\\\\
23
<> object FastLineForm:
object rastliner of m.

# <u>AC.&P</u>

Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clRed
VertAxis = aRightAxis
LinePen.Color = clRed
LinePen.Style = psDot
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.0000000000000000000
YValues.Order = loNone
Left = 199
Top = 204
end
<> object FastLineSeries2:
TFastLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clGreen
LinePen.Color = clGreen
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.0000000000000000000000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000000000000000000000
YValues.Order = loNone
Left = 237
Top = 206
end
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 153
Height = 381
Align = alLeft
TabOrder = 1
<> object Button1: TButton
Left = 13

<u>.                                    </u>
13
BottomAxis.LabelsFont.Name =
'Arial'
BottomAxis.LabelsFont.Style = []
BottomAxis.Ticks.Color = clLime
ClipPoints = False
Frame.Color = clBlue
Frame.Visible = False
LeftAxis.Grid.Color = clBlue
LeftAxis.LabelsFont.Charset =
DEFAULT CHARSET
LeftAxis.LabelsFont.Color =
clNavy
LeftAxis.LabelsFont.Height = -13
LeftAxis.LabelsFont.Name =
LettAxis.LabeisFont.Name = 'Arial'
LeftAxis.LabelsFont.Style = []
LeftAxis.Ticks.Color = clRed
Legend.Visible = False
RightAxis.Grid.Visible = False
RightAxis.LabelsFont.Charset =
DEFAULT_CHARSET
RightAxis.LabelsFont.Color =
clRed
RightAxis.LabelsFont.Height = -
13
RightAxis.LabelsFont.Name =
'Arial'
RightAxis.LabelsFont.Style = []
RightAxis.Ticks.Color = clBlue
TopAxis.Grid.Visible = False
TopAxis.LabelsFont.Charset =
DEFAULT_CHARSET
TopAxis.LabelsFont.Color =
clGreen
TopAxis.LabelsFont.Height = -13
TopAxis.LabelsFont.Name =
'Arial'
TopAxis.LabelsFont.Style = []
TopAxis.Ticks.Color = clYellow
View3D = False
Align = alClient
BevelOuter = bvNone
BevelWidth = 6
Color = clSilver
TabOrder = 0
<pre>&lt;&gt; object FastLineSeries1:</pre>
TFastLineSeries
HorizAxis = aTopAxis

77.13.
Height = 33
TabOrder = 4
Kind = bkClose
end
<> object Memo1: TMemo
Left = 12
Top = 144
Width = 121
Height = 181
Lines.Strings = (
'Use this demo to'
'benchmark your'
'video and system '
'speed.'
"
'Buffered display '
'avoids flickering and '
'can improve '
'performance on modern '
'hardware.'
naruware.
IFag4I in a Coning and
'FastLine Series are'
'faster than regular'
'Line Series.')
TabOrder = 5
end
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
24
<> object FeaturesForm:
TFeaturesForm
Left = 186
Top = 108
BorderIcons = [biSystemMenu]
BorderStyle = bsDialog
Caption = 'TeeChart Features.
100% Native VCL / CLX'
ClientHeight = 479
ClientWidth = 609
Color = clNavy
Font.Charset =
DEFAULT CHARSET
Font.Color = clWhite
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = [] OldCreateOrder = True
regiet roer = 1 roe

Top = 8
Width = 128
Height = 33
Caption = '&Speed Test !'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -16
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 0
OnClick = Button1Click
end
<> object CheckBox1:
TCheckBox
Left = 26
Top = 52
Width = 97
Height = 17
Caption = 'Buffered Display'
TabOrder = 1
OnClick = CheckBox1Click
end
<> object CheckBox2: TCheckBox
Left = 26
Top = 76
Width = 79
Height = 17
Caption = 'Draw &Axis'
TabOrder = 2
OnClick = CheckBox2Click
end
<> object CheckBox3:
TCheckBox
Left = 26
Top = 100
Width = 75
Height = 17
Caption = '&Clip Points'
TabOrder = 3
OnClick = CheckBox3Click
end
<> object BitBtn3: TBitBtn
Left = 28
Top = 340
Width = 89

Left = 104
Top = 76
Width = 109
Height = 16
HelpType = htKeyword
Caption = 'Animated Zoom'
Color = clSilver
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
Transparent = True
end
<> object Label23: TLabel
Left = 104
Top = 111
Width = 146
Height = 16
HelpType = htKeyword
Caption = 'BackGround
Bitmaps'
Color = clSilver
Font.Charset =
DEFAULT_CHARSET
DEFAULT_CHARSET Font.Color = clBlack
DEFAULT_CHARSET
DEFAULT_CHARSET Font.Color = clBlack
DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -13
DEFAULT_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif'
DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -13  Font.Name = 'MS Sans Serif'  Font.Style = [fsBold]
DEFAULT_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentColor = False ParentFont = False
DEFAULT_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentColor = False ParentFont = False Transparent = True
DEFAULT_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentColor = False ParentFont = False Transparent = True end
DEFAULT_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentColor = False ParentFont = False Transparent = True end <> object Label24: TLabel
DEFAULT_CHARSET  Font.Color = clBlack  Font.Height = -13  Font.Name = 'MS Sans Serif'  Font.Style = [fsBold]  ParentColor = False  ParentFont = False  Transparent = True end  <> object Label24: TLabel  Left = 104
DEFAULT_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentColor = False ParentFont = False Transparent = True end <> object Label24: TLabel Left = 104 Top = 147
DEFAULT_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentColor = False ParentFont = False Transparent = True end <> object Label24: TLabel Left = 104 Top = 147 Width = 112
DEFAULT_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentColor = False ParentFont = False Transparent = True end <> object Label24: TLabel Left = 104 Top = 147 Width = 112 Height = 16
DEFAULT_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentColor = False ParentFont = False Transparent = True end <> object Label24: TLabel Left = 104 Top = 147 Width = 112 Height = 16 HelpType = htKeyword
DEFAULT_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentColor = False ParentFont = False Transparent = True end <> object Label24: TLabel Left = 104 Top = 147 Width = 112 Height = 16 HelpType = htKeyword Caption = 'Custom Drawing'
DEFAULT_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentColor = False ParentFont = False Transparent = True end <> object Label24: TLabel Left = 104 Top = 147 Width = 112 Height = 16 HelpType = htKeyword Caption = 'Custom Drawing' Color = clSilver
DEFAULT_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentColor = False ParentFont = False Transparent = True end <> object Label24: TLabel Left = 104 Top = 147 Width = 112 Height = 16 HelpType = htKeyword Caption = 'Custom Drawing' Color = clSilver Font.Charset =
DEFAULT_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentColor = False ParentFont = False Transparent = True end <> object Label24: TLabel Left = 104 Top = 147 Width = 112 Height = 16 HelpType = htKeyword Caption = 'Custom Drawing' Color = clSilver Font.Charset = DEFAULT_CHARSET
DEFAULT_CHARSET Font.Color = clBlack Font.Height = -13 Font.Name = 'MS Sans Serif' Font.Style = [fsBold] ParentColor = False ParentFont = False Transparent = True end <> object Label24: TLabel Left = 104 Top = 147 Width = 112 Height = 16 HelpType = htKeyword Caption = 'Custom Drawing' Color = clSilver Font.Charset =

Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Label14: TLabel
Left = 16
Top = 457
Width = 300
Height = 13
HelpType = htKeyword
Caption = '© Copyright 1996-2001
by David Berneda. All Rights
Reserved.'
Color = clNavy
Font.Charset =
DEFAULT_CHARSET
Font.Color = clSilver
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentColor = False
ParentFont = False
end
<> object Panel2: TPanel
Left = 16
Top = 16
Width = 569
Height = 417
TabOrder = 20
<> object Label21: TLabel
Left = 104
Top = 40
Width = 140
Height = 16
HelpType = htKeyword
Caption = 'Custom Point Colors'
Color = clSilver
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
Transparent = True
end
<> object Label22: TLabel

Caption = 'Overlayed Bars'
Color = clSilver
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
Transparent = True
end
<> object Label28: TLabel
Left = 104
Top = 289
Width = 131
Height = 16
HelpType = htKeyword
Caption = 'Controlling Legend'
Color = clSilver
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
Transparent = True
end
<> object Label29: TLabel
Left = 104
Top = 324
Width = 153
Height = 16
HelpType = htKeyword
Caption = 'Custom Axis
Labelling'
Color = clSilver
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
Transparent = True

Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
Transparent = True
end
<> object Label25: TLabel
Left = 104
Top = 182
Width = 108
Height = 16
HelpType = htKeyword
Caption = 'Custom Legend'
Color = clSilver
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
Transparent = True
end
<> object Label26: TLabel
Left = 104
Top = 218
Width = 120
Height = 16
HelpType = htKeyword
Caption = 'Mini Small Charts'
Color = clSilver
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
Transparent = True
end
<> object Label27: TLabel
Left = 104
Top = 253
Width = 109
Height = 16
11019110 10
HelpType = htKeyword

Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
Transparent = True
end
<> object Label33: TLabel
Left = 384
Top = 111
Width = 77
Height = 16
HelpType = htKeyword
Caption = 'Scroll Bars'
Color = clSilver
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
Transparent = True
end
<> object Label34: TLabel
Left = 384
Top = 147
Width = 94
Height = 16
8
HelpType = htKeyword
Caption = 'Digital Charts'
Color = clSilver
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
Transparent = True
end
<> object Label35: TLabel
Left = 384
Top = 182
<b>Width = 74</b>

1
end
<> object Label30: TLabel
Left = 104
Top = 360
Width = 145
Height = 16
HelpType = htKeyword
Caption = 'Custom Axis
Drawing'
Color = clSilver
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
Transparent = True
end
<> object Label31: TLabel
Left = 384
Top = 40
Width = 117
Height = 16
HelpType = htKeyword
Caption = 'Custom Scrolling'
Color = clSilver
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
Transparent = True
end
<> object Label32: TLabel
Left = 384
Top = 76
Width = 122
Height = 16
HelpType = htKeyword
Caption = 'TeeChart Shapes'
Color = clSilver
Font.Charset =
DEFAULT CHARSET
DETAULI_CHARSEI

Transparent = True
end
<> object Label38: TLabel
Left = 384
Top = 289
Width = 114
Height = 16
HelpType = htKeyword
Caption = 'Logarithmic Axis'
Color = clSilver
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
Transparent = True
end
<> object Label39: TLabel
Left = 384
Top = 324
Width = 126
Height = 16
HelpType = htKeyword
Caption = 'Chart Auto-Paging'
Color = clSilver
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
Transparent = True
end
<> object Label40: TLabel
Left = 384
Top = 360
Width = 117
Height = 16
HelpType = htKeyword
Caption = 'Metafiles *.WMF '
Color = clSilver
Font.Charset =
DEFAULT_CHARSET

Height = 16
HelpType = htKeyword
Caption = 'Cross Hair'
Color = clSilver
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
Transparent = True
end
<> object Label36: TLabel
Left = 384
Top = 218
Width = 133
Height = 16
HelpType = htKeyword
Caption = 'Keyboard Scrolling'
Color = clSilver
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
Transparent = True
end
<> object Label37: TLabel
Left = 384
Top = 253
Width = 108
Height = 16
HelpType = htKeyword
Caption = 'Custom Printing'
Color = clSilver
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False

•
end
<> object BitBtn9: TBitBtn
Left = 40
Top = 261
Width = 65
Height = 27
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 4
OnClick = BitBtn9Click
Glyph.Data = {
end
<> object BitBtn11: TBitBtn
Left = 312
Top = 332
Width = 65
Height = 27
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 5
OnClick = BitBtn11Click
Glyph.Data = {
NumGlyphs = 2
end
<> object BitBtn10: TBitBtn
Left = 312
Top = 261
Width = 65
Height = 27
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 6
OnClick = BitBtn10Click
Onence Didding ones

,
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
Transparent = True
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 569
Height = 17
Caption = 'Choose a feature !'
Color = clYellow
Font.Charset =
DEFAULT CHARSET
Font.Color = clNavy
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 0
end
end
<> object BitBtn5: TBitBtn
Left = 40
Top = 190
Width = 65
Height = 27
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 0
OnClick = BitBtn5Click
Glyph.Data = {
000333333333333F7773333333333333
00033333333333F777333333333333
00
03333333FFF3F777333333700073B
70333333F7773F7773333330777770
0B3
NumGlyphs = 2
V 1

TabOrder = 9
OnClick = BitBtn12Click
Glyph.Data = {
end
<> object BitBtn13: TBitBtn
Left = 40
Top = 297
Width = 65
Height = 27
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 10
OnClick = BitBtn13Click
Glyph.Data = {
777770000000}
end
<> object BitBtn14: TBitBtn
Left = 312
Top = 155
Width = 65
Height = 27
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 11
OnClick = BitBtn14Click
Glyph.Data = {
end
<> object BitBtn15: TBitBtn
Left = 312
Top = 368
Width = 65
Height = 27
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = []
V 13

Glyph.Data = {
NumGlyphs = 2
end
<> object BitBtn2: TBitBtn
Left = 312
Top = 48
Width = 65
Height = 27
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 7
OnClick = BitBtn2Click
Glyph.Data = {
end
<> object BitBtn8: TBitBtn
Left = 312
Top = 84
Width = 65
Height = 27
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 8
OnClick = BitBtn8Click
Glyph.Data = {
777777777777770000}
end
<> object BitBtn12: TBitBtn
Left = 40
Top = 332 Width = 65
Height = 27
Font.Charset = DEFAULT CHARSET
<del>-</del>
Font Height = 13
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False

E 4 C4 L D
Font.Style = []
ParentFont = False
TabOrder = 16
OnClick = BitBtn18Click
Glyph.Data = {
end
<> object BitBtn19: TBitBtn
Left = 312
Top = 190
Width = 65
Height = 27
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 17
OnClick = BitBtn19Click
Glyph.Data = {
NumGlyphs = 2
end
<> object BitBtn20: TBitBtn
Left = 312
Top = 226
Width = 65
Height = 27
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 18
OnClick = BitBtn20Click
Glyph.Data = {
NumGlyphs = 2
end
<> object BitBtn21: TBitBtn
Left = 312
Top = 297
Width = 65
Height = 27
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack

ParentFont = False
TabOrder = 13
OnClick = BitBtn15Click
Glyph.Data = {
end
<> object BitBtn16: TBitBtn
Left = 40
Top = 368
Width = 65
Height = 27
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 14
OnClick = BitBtn16Click
Glyph.Data = {
end
<> object BitBtn17: TBitBtn
Left = 40
Top = 48
Width = 65
Height = 27
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 15
OnClick = BitBtn17Click
Glyph.Data = {
8888888888888880000}
end
<> object BitBtn18: TBitBtn
Left = 312
Top = 119
Width = 65
Height = 27
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
I Ullus famic 1910 Dalla Del II

Haight = 277	Font Hoight - 12
Height = 377	Font.Height = -13
AnimatedZoom = True	Font.Name = 'MS Sans Serif'
AnimatedZoomSteps = 4	Font.Style = []
BackWall.Brush.Color = clWhite	ParentFont = False
BackWall.Brush.Style = bsClear	TabOrder = 19
BackWall.Color = clSilver	OnClick = BitBtn21Click
Foot.Font.Charset =	Glyph.Data = {
DEFAULT_CHARSET	<> object BitBtn22: TBitBtn
Foot.Font.Color = clRed	Left = 492
Foot.Font.Height = -12	Top = 440
Foot.Font.Name = 'Arial'	Width = 95
Foot.Font.Style = [fsItalic]	Height = 31
Foot.Frame.Color = clScrollBar	Caption = 'Close'
Gradient.EndColor = clWhite	Font.Charset =
Gradient.StartColor = clYellow	DEFAULT_CHARSET
Title.Alignment = taLeftJustify	Font.Color = clBlack
Title.Brush.Color = 8454143	Font.Height = -11
Title.Color = 8454143	Font.Name = 'MS Sans Serif'
Title.Font.Charset =	Font.Style = []
DEFAULT_CHARSET	ParentFont = False
Title.Font.Color = clRed	TabOrder = 21
Title.Font.Height = -19	Kind = bkClose
Title.Font.Name = 'Arial'	end
Title.Font.Style = [fsBold, fsItalic]	End
Title.Frame.Color = clGreen	
Title.Frame.Width = 4	111111111111111111111111111111111111111
Title.Text.Strings = (	25
'Zoom by clicking Left Mouse	<> object
button'	FormAnimatedZoom:
'and dragging bottom right.')	TFormAnimatedZoom
BackColor = clSilver	Left = 197
BottomAxis.Grid.Color =	Top = 113
clScrollBar	Width = 619
Chart3DPercent = 46	Height = 411
LeftAxis.Grid.Color = clScrollBar	ActiveControl = CheckBox1
Legend.Visible = False	Caption = 'TeeChart Animated
RightAxis.Grid.Color =	Zoom Example'
clScrollBar	Color = clBtnFace
TopAxis.Grid.Color = clScrollBar	ParentFont = True
View3D = False	OldCreateOrder = True
Align = alClient	Position = poScreenCenter
Color = clWhite	OnCreate = FormCreate
TabOrder = 0	PixelsPerInch = 96
<> object LineSeries1:	TextHeight = 13
TLineSeries	<> object Chart1: TChart
Marks.ArrowLength = 8	Left = 141
Marks.Visible = False	Top = 0
SeriesColor = clRed	Width = 470
SCHOOLINE CHACU	Wiuii — 4/0

XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000000000000000000000
YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000000000000000000000
YValues.Multiplier = 1.000000000000000000000000000000000000
1.000000000000000000000000000000000000
1.000000000000000000000000000000000000
Left = 244 Top = 100 end <> object LineSeries4: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clBlue Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Multiplier = 1.000000000000000000000000000000000000
Left = 244 Top = 100 end <> object LineSeries4: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clBlue Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Multiplier = 1.000000000000000000000000000000000000
Top = 100 end  <> object LineSeries4: TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False SeriesColor = clBlue Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.00000000000000000 XValues.Order = loAscending YValues.Name = 'Y' YValues.Multiplier = 1.00000000000000000 YValues.Order = loNone Left = 276 Top = 100 end <> object LineSeries5: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clWhite Pointer.InflateMargins = True
end <> object LineSeries4: TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False SeriesColor = clBlue Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000  XValues.Order = loAscending YValues.Name = 'Y' YValues.Multiplier = 1.00000000000000000  YValues.Order = loNone Left = 276 Top = 100 end <> object LineSeries5: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clWhite Pointer.InflateMargins = True
<pre>&lt;&gt; object LineSeries4: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clBlue Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000</pre>
TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clBlue  Pointer.InflateMargins = True  Pointer.Style = psRectangle  Pointer.Visible = False  XValues.DateTime = False  XValues.Name = 'X'  XValues.Multiplier =  1.00000000000000000000000000000000000
Marks.ArrowLength = 8  Marks.Visible = False SeriesColor = clBlue Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.0000000000000000000 XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.00000000000000000 YValues.Order = loNone Left = 276 Top = 100 end <> object LineSeries5: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clWhite Pointer.InflateMargins = True
Marks.Visible = False SeriesColor = clBlue Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000 XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.00000000000000000 YValues.Order = loNone Left = 276 Top = 100 end <> object LineSeries5: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clWhite Pointer.InflateMargins = True
SeriesColor = clBlue Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000 XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.00000000000000000 YValues.Order = loNone Left = 276 Top = 100 end <> object LineSeries5: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clWhite Pointer.InflateMargins = True
Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000 XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000 YValues.Order = loNone Left = 276 Top = 100 end <> object LineSeries5: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clWhite Pointer.InflateMargins = True
Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000
Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000 XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.00000000000000000 YValues.Order = loNone Left = 276 Top = 100 end <> object LineSeries5: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clWhite Pointer.InflateMargins = True
XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000 XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.00000000000000000 YValues.Order = loNone Left = 276 Top = 100 end <> object LineSeries5: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clWhite Pointer.InflateMargins = True
XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000
XValues.Multiplier = 1.000000000000000000000000000000000000
1.00000000000000000  XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.00000000000000000  YValues.Order = loNone Left = 276 Top = 100 end <> object LineSeries5: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clWhite Pointer.InflateMargins = True
XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000 YValues.Order = loNone Left = 276 Top = 100 end <> object LineSeries5: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clWhite Pointer.InflateMargins = True
YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000 YValues.Order = loNone Left = 276 Top = 100 end <> object LineSeries5: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clWhite Pointer.InflateMargins = True
YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000000  YValues.Order = loNone Left = 276 Top = 100 end <> object LineSeries5: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clWhite Pointer.InflateMargins = True
YValues.Multiplier = 1.000000000000000000000000000000000000
1.0000000000000000000000  YValues.Order = loNone  Left = 276  Top = 100  end  <> object LineSeries5:  TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clWhite  Pointer.InflateMargins = True
YValues.Order = loNone  Left = 276  Top = 100  end  <> object LineSeries5:  TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clWhite  Pointer.InflateMargins = True
Left = 276 Top = 100 end <> object LineSeries5: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clWhite Pointer.InflateMargins = True
Top = 100 end <> object LineSeries5: TLineSeries Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clWhite Pointer.InflateMargins = True
end <> object LineSeries5: TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False SeriesColor = clWhite Pointer.InflateMargins = True
end <> object LineSeries5: TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False SeriesColor = clWhite Pointer.InflateMargins = True
<> object LineSeries5: TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False SeriesColor = clWhite Pointer.InflateMargins = True
TLineSeries  Marks.ArrowLength = 8  Marks.Visible = False  SeriesColor = clWhite  Pointer.InflateMargins = True
Marks.ArrowLength = 8 Marks.Visible = False SeriesColor = clWhite Pointer.InflateMargins = True
Marks.Visible = False SeriesColor = clWhite Pointer.InflateMargins = True
SeriesColor = clWhite Pointer.InflateMargins = True
Pointer.InflateMargins = True
Pointon Stylo — ngDogtonglo
Pointer.Style = psRectangle Pointer.Visible = False
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
X7X7 1 X # 14 10
YValues.Multiplier =
1.0000000000000000000

Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000000000000000000000
YValues.Order = loNone
Left = 172
Top = 100
end
<> object LineSeries2:
TLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clGreen
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier = 1.00000000000000000000000000000000000
YValues.Order = loNone
Left = 208
Top = 100
end
<> object LineSeries3:
TLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clYellow
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = False
XValues.Name = 'X'
XValues.Nultiplier =
<u> </u>
1.000000000000000000

CL L T
Checked = True
State = cbChecked
TabOrder = 0
OnClick = CheckBox1Click
end
<> object CheckBox2:
TCheckBox
Left = 15
Top = 12
Width = 42
Height = 17
Caption = '3&D'
TabOrder = 1
OnClick = CheckBox2Click
end
<> object SpinEdit1:
TSpinEdit
Left = 86
Top = 64
Width = 47
Height = 22
MaxValue = 1000
MinValue = 1
TabOrder = 2
Value = 1
OnChange = SpinEdit1Change
end
<> object BitBtn2: TBitBtn
Left = 26
Top = 140
Width = 89
Height = 33
3
Caption = 'Zoom OUT'
TabOrder = 3
OnClick = BitBtn2Click
Glyph.Data = {
end
<> object BitBtn3: TBitBtn
Left = 26
Top = 344
Width = 89
Height = 33
TabOrder = 5
OnClick = BitBtn2Click
Kind = bkClose
end
<> object Memo1: TMemo
Left = 12
LCIL 12

T 400
Top = 100
end
<> object PointSeries1:
TPointSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clGray
Pointer.Brush.Color = 4259584
Pointer.InflateMargins = True
Pointer.Style = psDiamond
Pointer.Visible = True
XValues.DateTime = False
XValues.Name = 'X'
XValues.Nultiplier =
1.000000000000000000000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 348
Top = 100
end
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 141
Height = 377
Align = alLeft
TabOrder = 1
<> object Label1: TLabel
Left = 14
Top = 68
Width = 60
Height = 13
HelpType = htKeyword
Caption = 'Zoom &Steps:'
FocusControl = SpinEdit1
end
<> object CheckBox1:
TCheckBox
Left = 15
Top = 32
Width = 110
Height = 17
Caption = '&Animated Zoom !!!'
Capuon & Anninateu Loom :::

LeftWall.Color = clWhite	Top = 192
Title.Alignment = taLeftJustify	Width = 121
Title.Brush.Color = clBlack	Height = 129
Title.Color = clBlack	Lines.Strings =
Title.Font.Charset =	'Animated Zoo
DEFAULT CHARSET	'can be activat
Title.Font.Color = clYellow	'to better inde
Title.Font.Height = -16	'zoomed areas.
Title.Font.Name = 'Arial'	"
Title.Font.Style = [fsBold]	'Try to zoom u
Title.Frame.Color = clBlue	'the left mouse
Title.Frame.Width = 4	'with and with
Title.Frame.Visible = True	'Animated Zoo
Title.Text.Strings = (	TabOrder = 6
'Average, High and Low.')	end
BottomAxis.Axis.Color = clRed	end
BottomAxis.DateTimeFormat =	End
'dd/MMM'	\\\\\\\\
BottomAxis.Grid.Color = clBlack	26
BottomAxis.Grid.Style = psSolid	<> object Hig
BottomAxis.LabelsFont.Charset =	THighLowForm
DEFAULT_CHARSET	Left = 191
BottomAxis.LabelsFont.Color =	Top = 110
clPurple	Width = 616
BottomAxis.LabelsFont.Height = -	Height = 440
11	Caption = 'TeeCh
BottomAxis.LabelsFont.Name =	Demo'
'MS Sans Serif'	Color = clBtnFace
BottomAxis.LabelsFont.Style = []	ParentFont = True
BottomAxis.MinorTickCount = 0	OldCreateOrder =
BottomAxis.TickLength = 9	Position = poScree
BottomAxis.Ticks.Color = clBlue	OnCreate = Form
Chart3DPercent = 45	PixelsPerInch = 90
Frame.Width = 3	TextHeight = 13
LeftAxis.Axis.Color = clBlue	<> object Ch
LeftAxis.LabelsFont.Charset =	Left = 153
DEFAULT_CHARSET	Top = 0
LeftAxis.LabelsFont.Color =	<b>Width = 455</b>
clRed	Height = 406
LeftAxis.LabelsFont.Height = -13	BackWall.Brush.
LeftAxis.LabelsFont.Name =	BackWall.Brush.
'Arial'	BackWall.Pen.W
LeftAxis.LabelsFont.Style =	Foot.Font.Charse
[fsBold]	DEFAULT_CHAR
Legend.ShadowColor = clGray	Foot.Font.Color
Legend.ShadowSize = 4	Foot.Font.Height
RightAxis.Automatic = False	Foot.Font.Name
RightAxis.AutomaticMaximum =	Foot.Font.Style =
False	

Top = 192
Width = 121
Height = 129
Lines.Strings = (
'Animated Zoom'
'can be activated'
'to better indentify'
'zoomed areas.'
11
'Try to zoom using'
'the left mouse button,'
'with and without'
'Animated Zoom !')
TabOrder = 6
end
end
End
\\\\\\\\\
26
<> object HighLowForm:
THighLowForm
Left = 191
Top = 110
Width = 616
Height = 440
Caption = 'TeeChart Statistics
Demo'
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Chart1: TChart
Left = 153
Top = 0
Width = 455
Height = 406
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Pen.Width = 3
Foot.Font.Charset =
DEFAULT CHARSET
Foot.Font.Color = clRed
Foot.Font.Height = -12
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsItalic]
· · · · · · · · · · · · · · · · · · ·

Pointer.Visible = True  XValues.DateTime = True  XValues.Name = 'X'  XValues.Multiplier =  1.00000000000000000  XValues.Order = loAscending  YValues.DateTime = False  YValues.Name = 'Y'  YValues.Multiplier =  RightAxis.AutomaticMinimum False  RightAxis.LabelsFont.Charset =  DEFAULT_CHARSET  RightAxis.LabelsFont.Color =  clBlack  RightAxis.LabelsFont.Height =  19
XValues.Name = 'X'  XValues.Multiplier = 1.00000000000000000  XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y'  YValues.Multiplier =  RightAxis.Grid.Visible = False RightAxis.LabelsFont.Charset = DEFAULT_CHARSET RightAxis.LabelsFont.Color = clBlack RightAxis.LabelsFont.Height = 19
XValues.Multiplier = 1.00000000000000000000000000000000000
1.000000000000000000000000000000000000
XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier =  RightAxis.LabelsFont.Color = clBlack RightAxis.LabelsFont.Height = 19
YValues.DateTime = False YValues.Name = 'Y' RightAxis.LabelsFont.Height = 19
YValues.Name = 'Y'  YValues.Multiplier = RightAxis.LabelsFont.Height = 19
YValues.Multiplier = 19
1 / wiwesit/initipiter
D' LA ' T L LE AN
1.0000000000000000000000 RightAxis.LabelsFont.Name =
YValues.Order = loNone 'Arial'
YValues.ValueSource = 'Y' RightAxis.LabelsFont.Style =
<> object [fsBold, fsItalic]
AverageSeriesAverage: OnAfterDraw = Chart1AfterDr
TAverageTeeFunction Align = alClient
end BevelInner = bvRaised
end BevelOuter = bvLowered
<> object HighSeries:  BorderWidth = 25
TLineSeries Color = 16777088
Marks.ArrowLength = 8 TabOrder = 0
Marks.Visible = False <> object BarSeries1:
DataSource = BarSeries1 TBarSeries
SeriesColor = clFuchsia Marks.ArrowLength = 20
Title = 'High' Marks.Visible = True
LinePen.Width = 2 SeriesColor = clRed
Pointer.Brush.Color = 8388863 BarStyle = bsPyramid
Pointer.InflateMargins = False XValues.DateTime = True
Pointer.Pen.Color = clBlue XValues.Name = 'X'
Pointer.Pen.Style = psDot XValues.Multiplier =
Pointer.Style = psRectangle 1.000000000000000000000000000000000000
Pointer.Visible = True XValues.Order = loAscending
XValues.DateTime = True YValues.DateTime = False
XValues.Name = 'Y'  YValues.Name = 'Y'
XValues.Multiplier = YValues.Multiplier =
1.0000000000000000000000000000000000000
XValues.Order = loAscending YValues.Order = loNone
YValues.DateTime = False Left = 47
YValues.Name = 'Y' Top = 117
YValues.Multiplier = end
1.00000000000000000 <> object AverageSeries:
YValues.Order = loNone TLineSeries
<> object HighSeriesHigh: Marks.ArrowLength = 8
THighTeeFunction Marks.Visible = False
end DataSource = BarSeries1
end SeriesColor = clGreen
<> object LowSeries: Title = 'Average'
TLineSeries LinePen.Width = 2
Marks.ArrowLength = 8 Pointer.InflateMargins = False
Marks.Visible = False Pointer.Style = psRectangle

DataSource = BarSeries1
SeriesColor = clBlue
Title = 'Low'
LinePen.Width = 2
Pointer.InflateMargins = False
Pointer.Style = psRectangle
Pointer.Visible = True
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.0000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.0000000000000000000
YValues.Order = loNone
<> object LowSeriesLow:
TLowTeeFunction
end
end
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 153
Height = 406
Align = alLeft
TabOrder = 1
<> object BitBtn1: TBitBtn
Left = 25
Top = 300
Width = 89
Height = 33
Caption = '&Print'
TabOrder = 0
OnClick = BitBtn1Click
end
<> object BitBtn2: TBitBtn
Left = 26
Top = 344
Width = 89
Height = 33
TabOrder = 1
Kind = bkClose
end
<> object CheckBox1:
TCheckBox

BackWall.Brush.Style = bsClear	Left = 12	
Foot.Alignment = taRightJustify	Top = 124	
Foot.Font.Charset =	Width = 121	
DEFAULT_CHARSET	Height = 161	
Foot.Font.Color = clYellow	Lines.Strings = (	
Foot.Font.Height = -12	'Statistical functions'	
Foot.Font.Name = 'Arial'	'can be applied to'	
Foot.Font.Style = [fsItalic]	'existing Series.'	
Foot.Frame.Color = clScrollBar	"	
Foot.Text.Strings = (	'This demo shows '	
	'Average, High and Low'	
'Left drag (Up/Down) to Zoom.	'functions on one or two'	
Right drag to Scroll. Invert Zoom r'	'normal Bar Series.')	
+	TabOrder = 6	
'ectangle to reset.')	end	
Title.Font.Charset =	end	
DEFAULT_CHARSET	<> object Timer1: TTimer	
Title.Font.Color = clWhite	Enabled = False	
Title.Font.Height = -16	Interval = 100	
Title.Font.Name = 'Arial'	OnTimer = Timer1Timer	
Title.Font.Style = [fsBold]	Left = 224	
Title.Frame.Color = clScrollBar	<b>Top = 6</b>	
Title.Text.Strings = (	end	
'This demo shows the Gantt	End	
chart Type.'		
'Each Point is a Horizontal Bar	27	
that defines a Date Period.'	<> object GanttForm:	
")	TGanttForm	
BottomAxis.LabelsAngle = 90	Left = 147	
BottomAxis.LabelsFont.Charset =	Top = 108	
DEFAULT CHARSET	Width = 686	
BottomAxis.LabelsFont.Color =	Height = 432	
clWhite	Caption = 'TeeChart Gantt Series	
BottomAxis.LabelsFont.Height = -	Demo'	
12	Color = clBtnFace	
BottomAxis.LabelsFont.Name =	ParentFont = True	
'Arial'	OldCreateOrder = True	
BottomAxis.LabelsFont.Style = []	Position = poScreenCenter	
BottomAxis.LabelStyle = talValue	OnCreate = FormCreate	
BottomAxis.MinorTicks.Color =	PixelsPerInch = 96	
clWhite	TextHeight = 13	
BottomAxis.TickInnerLength = 4	ĕ	
BottomAxis.Ticks.Color = clLime	<> object Chart1: TChart  Left = 137	
BottomAxis.TicksInner.Color =		
8388863	Top = 0	
BottomAxis.Title.Caption =	Width = 541	
'Current Year'	Height = 398	
BottomAxis.Title.Font.Charset =	AllowPanning = pmHorizontal	
DEFAULT CHARSET	BackWall.Brush.Color = clWhite	

Marks.Visible = True
SeriesColor = clRed
OnClick = GanttSeries1Click
Pointer.Draw3D = False
Pointer.HorizSize = 83
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.VertSize = 10
Pointer.Visible = True
XValues.DateTime = True
XValues.Name = 'Start'
XValues.Multiplier =
1.0000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.00000000000000000
YValues.Order = loNone
ConnectingPen.Color = clRed
ConnectingPen.Width = 2
StartValues.DateTime = True
StartValues.Name = 'Start'
StartValues.Multiplier =
1.000000000000000000000000000000000000
StartValues.Order = loAscending
EndValues.DateTime = False
EndValues.Name = 'End'
End Values. Nultiplier =
1.0000000000000000000000000000000000000
EndValues.Order = loNone
NextTask.DateTime = False
NextTask.Name = 'NextTask'
NextTask.Multiplier = 1.00000000000000000000000000000000000
NextTask.Order = loNone
end
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 137
Height = 398
Align = alLeft
TabOrder = 1
<> object Label1: TLabel
Left = 15
Top = 11

BottomAxis.Title.Font.Color = clFuchsia
BottomAxis.Title.Font.Height = -
19
BottomAxis.Title.Font.Name =
'Arial'
BottomAxis.Title.Font.Style = [fsItalic]
LeftAxis.Grid.Color = clLime
LeftAxis.LabelsFont.Charset =
DEFAULT_CHARSET
LeftAxis.LabelsFont.Color =
clAqua
LeftAxis.LabelsFont.Height = -13 LeftAxis.LabelsFont.Name =
'Arial'
LeftAxis.LabelsFont.Style = []
LeftAxis.LabelsSeparation = 3
LeftAxis.LabelStyle = talText
LeftAxis.Title.Font.Charset =
DEFAULT_CHARSET
LeftAxis.Title.Font.Color =
clPurple LeftAxis.Title.Font.Height = -16
LeftAxis.Title.Font.Name = 'Arial'
LeftAxis.Title.Font.Style =
[fsBold, fsItalic]
Legend.Visible = False
View3D = False
OnGetNextAxisLabel =
Chart1GetNextAxisLabel
Align = alClient Color = 8421440
TabOrder = 0
OnMouseMove =
Chart1MouseMove
<> object GanttSeries1:
TGanttSeries
ColorEachPoint = True
Marks.ArrowLength = 0
Marks.BackColor = clWhite Marks.Clip = True
Marks.Cnp = 1rue Marks.Font.Charset =
DEFAULT CHARSET
Marks.Font.Color = clNavy
Marks.Font.Height = -11
Marks.Font.Name = 'Arial'
Marks.Font.Style = []
Marks.Frame.Color = clRed

'Gantt Series '
'are made of points'
'with Start and End'
'coordinates.'
11
'This example shows'
'how to get mouse '
'position and how'
'to customize bottom'
'axis labels.')
TabOrder = 1
end
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
28
<> object KeyboardForm:
TKeyboardForm
Left = 190
Top = 116
Width = 667
Height = 389
ActiveControl = Chart1
Caption = 'TeeChart Keyboard
Scrolling Example'
Color = clBtnFace
ParentFont = True
KeyPreview = True
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
OnKeyDown = FormKeyDown
PixelsPerInch = 96
TextHeight = 13
<> object Chart1: TChart
Left = 157
Top = 0
Width = 502
Height = 355
AnimatedZoom = True
AnimatedZoomSteps = 3
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Color = clWhite
BottomWall.Color = 8454016
BottomWall.Size = 16
Foot.Font.Charset =
DEFAULT CHARSET
DEFAULT_CHARSET

Width = 62 Height = 24
HelpType = htKeyword
Caption = 'Label1'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -20
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
end
<> object Shape1: TShape
Left = 24
Top = 84 Width = 77
Height = 29
HelpType = htKeyword
end
<> object Label2: TLabel
Left = 15
Top = 39
Width = 62
Height = 24
HelpType = htKeyword
Caption = 'Label1'
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlue
Font.Height = -20
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
end
<> object BitBtn3: TBitBtn
Left = 24
Top = 304
Width = 89
Height = 33
TabOrder = 0
Kind = bkClose
end
<> object Memo1: TMemo
Left = 8
Top = 128
Width = 121
Height = 161
Lines.Strings = (

Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clRed
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.0000000000000000000000
YValues.Order = loNone
Left = 199
Top = 121
end
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 157
Height = 355
Align = alLeft
TabOrder = 1
<> object BitBtn1: TBitBtn
Left = 37
Top = 284
Width = 89
Height = 33
TabOrder = 0
Kind = bkClose
end
<> object InvertScroll:
TCheckBox
Left = 21
Top = 12
Width = 128
Height = 17
Caption = '&Inverted Scrolling'
TabOrder = 1
OnClick = InvertScrollClick
end
<> object CheckLimits: TCheckBox
Left = 21

<u>1                                    </u>
Foot.Font.Color = clRed
Foot.Font.Height = -12
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsItalic]
Foot.Frame.Color = clScrollBar
Foot.Text.Strings = (
'The Form.OnKeyDown event
does the job. See source code. ')
LeftWall.Size = 16
Title.AdjustFrame = False
Title.Brush.Color = 8453888
Title.Color = 8453888
Title.Font.Charset =
DEFAULT CHARSET
Title.Font.Color = clBlack
Title.Font.Height = -12
Title.Font.Name = 'Arial'
Title.Font.Style = []
Title.Frame.Color = 8388863
Title.Frame.Width = 3
Title.Frame.Visible = True
Title.Text.Strings = (
Titic. Text.Strings – (
'This example shows how to
handle keyboard events to scroll
TChar' +
't contents.'
t contents.
'Use the keyboard arrow keys
and page up / page down keys to
scr' +
'oll.'
'Press SPACE to reset. Press
SHIFT and arrow keys to ZOOM.')
BackColor = clWhite
BottomAxis.Grid.Color =
clScrollBar
Chart3DPercent = 20
LeftAxis.Grid.Color = clScrollBar
Legend.Visible = False
RightAxis.Grid.Color =
clScrollBar
TopAxis.Grid.Color = clScrollBar
Align = alClient
Color = 16777088
TabOrder = 0
TabStop = True
<> object LineSeries1:
· ·
TLineSeries

OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 16
<> object Label14: TLabel
Left = 17
Top = 393
Width = 300
Height = 13
HelpType = htKeyword
Caption = '© Copyright 1996-2001
by David Berneda. All Rights
Reserved.'
Color = clNavy
Font.Charset =
DEFAULT CHARSET
Font.Color = clSilver
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentColor = False
ParentFont = False
end
<> object GroupBox1:
TGroupBox
Left = 16
Top = 8
Width = 553
Height = 233
Color = clSilver
Ctl3D = True
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentCtl3D = False
ParentFont = False
TabOrder = 0
<> object Label1: TLabel
Left = 51
Top = 36
Width = 134
Height = 16
HelpType = htKeyword
Alignment = taRightJustify
Caption =

T. 22
Top = 32
Width = 97
Height = 17
Caption = 'Check &Limits'
TabOrder = 2
OnClick = CheckLimitsClick
end
<> object Memo1: TMemo
Left = 20
Top = 76
Width = 121
Height = 161
Lines.Strings = (
'Scrolling and zooming'
'can also be done using'
'the keyboard.'
!!
'This example shows '
'how to scroll and zoom'
'a Chart using the Form'
'OnKeyDown event.' '')
,
TabOrder = 3
end
end
end End
end End \\\\\\\
end End \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
end End \\\\\\\
end End \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
end End \\\\\\\\\ 29 <> object SeriesForm:
end End  \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
end End               29 <> object SeriesForm: TSeriesForm Left = 190 Top = 108
end End \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
end End \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
end End
end End
end End \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
end
end
end
end

_
Left = 136
Top = 155
Width = 48
Height = 16
HelpType = htKeyword
Alignment = taRightJustify
Caption = 'Arrows'
Color = clSilver
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
end
<> object Label5: TLabel
Left = 112
Top = 194
Width = 69
Height = 16
HelpType = htKeyword
Alignment = taRightJustify
Caption = 'Statistical'
Color = clSilver
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font Name = IMS Song Soviet
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
end
<> object Label6: TLabel
Left = 384
Top = 135
Width = 50
Height = 16
HelpType = htKeyword
Caption = '&Bubble'
end
<> object Label7: TLabel
Left = 384
Top = 93
Width = 37
Height = 16
_

'&Line,Point,Area,Bar'
Color = clSilver
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
end
<> object Label2: TLabel
Left = 156
Top = 76
Width = 24
Height = 16
HelpType = htKeyword
Alignment = taRightJustify
Caption = '& Pie'
Color = clSilver
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
end
<> object Label3: TLabel
Left = 88
Top = 115
Width = 94
Height = 16
HelpType = htKeyword
Alignment = taRightJustify
Caption = 'Stacked Bars'
Color = clSilver
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
end
<> object Label4: TLabel

OnClick = BitBtn5Click
Glyph.Data = {
end
<> object BitBtn6: TBitBtn
Left = 309
Top = 126
Width = 57
Height = 33
Font.Charset =
DEFAULT CHARSET
Font.Color = clNavy
Font.Height = -16
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 2
OnClick = BitBtn6Click
Glyph.Data = {
end
<> object BitBtn7: TBitBtn
Left = 309
Top = 84
Width = 57
Height = 33
Font.Charset =
DEFAULT CHARSET
Font.Color = clNavy
Font.Height = -16
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 3
OnClick = BitBtn7Click
Glyph.Data = {
end
<> object BStacked: TBitBtn
Left = 200
Top = 105
Width = 57
Height = 33
Font.Charset =
DEFAULT_CHARSET
Font.Color = clNavy
Font.Height = -16
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 4
1 ab 01 uci = <b>7</b>

HolnTyno - htVoyword
HelpType = htKeyword Caption = '&Gantt'
•
end
<> object Label8: TLabel
Left = 384
Top = 52
Width = 64
Height = 16
HelpType = htKeyword
Caption = '&Fast Line'
end
<> object Label9: TLabel
Left = 384
Top = 176
Width = 103
Height = 16
HelpType = htKeyword
Caption = 'Stacked Areas'
end
<> object BitBtn4: TBitBtn
Left = 200
Top = 27
Width = 57
Height = 33
Font.Charset =
DEFAULT CHARSET
Font.Color = clNavy
Font.Height = -16
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 0
OnClick = BitBtn4Click
Glyph.Data = {
end
<> object BitBtn5: TBitBtn
Left = 200
Top = 66
Width = 57
Height = 33
Font.Charset =
DEFAULT_CHARSET
Font.Color = clNavy
Font.Height = -16
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 1

Font.Style = []
ParentFont = False
TabOrder = 7
OnClick = BStackedAreasClick
Glyph.Data = {
end
<> object BitBtn21: TBitBtn
Left = 201
Top = 183
Width = 56
Height = 35
Font.Charset =
DEFAULT CHARSET
Font.Color = clNavy
Font.Height = -16
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 8
OnClick = BitBtn21Click
Glyph.Data = {
NumGlyphs = 2
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 553
Height = 17
Caption = 'Standard Chart
Types'
Color = clYellow
Font.Charset =
DEFAULT_CHARSET
Font.Color = clNavy
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 9
end
end
<> object GroupBox3:
TGroupBox
Left = 16
Top = 255
Width = 553
Height = 114
Color = clSilver
COIOI CIBIIYCI

OnClick = BStackedClick
Glyph.Data = {
end
<> object BFastLine:
TBitBtn
Left = 309
Top = 43
Width = 57
Height = 33
Font.Charset =
DEFAULT CHARSET
Font.Color = clNavy
Font.Height = -16
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 5
OnClick = BFastLineClick
Glyph.Data = {
end
<> object BArrowSeries:
TBitBtn
Left = 200
Top = 144
Width = 57
Height = 33
Font.Charset =
DEFAULT_CHARSET
Font.Color = clNavy
Font.Height = -16
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 6
OnClick = BArrowSeriesClick
Glyph.Data = {
end
<> object BStackedAreas:
TBitBtn
Left = 309
Top = 167
Width = 57
Height = 33
Font.Charset =
DEFAULT_CHARSET
Font.Color = clNavy
Font.Height = -16
Font.Name = 'MS Sans Serif'

77 . 67
Font.Charset =
DEFAULT_CHARSET
Font.Color = clNavy
Font.Height = -16
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 0
OnClick = BitBtn9Click
Glyph.Data = {
end
<> object BitBtn10: TBitBtn
Left = 200
Top = 68
Width = 57
Height = 33
Font.Charset =
DEFAULT CHARSET
Font.Color = clNavy
Font.Height = -16
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 1
OnClick = BitBtn10Click
Glyph.Data = {
end
<> object BitBtn11: TBitBtn
Left = 309
Top = 24 Width = 57
Height = 33
Font.Charset =
DEFAULT_CHARSET
Font.Color = clNavy
Font.Height = -16
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 2
OnClick = BitBtn11Click
Glyph.Data = {
end
<> object BitBtn14: TBitBtn
Left = 309
Top = 68
Width = 57
Height = 33

Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentColor = False
ParentFont = False
TabOrder = 1
<> object Label10: TLabel
Left = 82
Top = 36
Width = 103
Height = 16
HelpType = htKeyword
Alignment = taRightJustify
Caption = 'P&ie from Table'
end
<> object Label11: TLabel
Left = 74
Top = 76
Width = 111
Height = 16
HelpType = htKeyword
Alignment = taRightJustify
Caption = '&SQL Query Bars'
end
<> object Label12: TLabel
Left = 384
Top = 34
Width = 100
Height = 16
HelpType = htKeyword
Caption = 'Lin&ked Tables'
end
<> object Label13: TLabel
Left = 384
Top = 74
Width = 132
Height = 16
HelpType = htKeyword
Caption = 'Horizontal DB Bars'
end
<> object BitBtn9: TBitBtn
Left = 200
Top = 24
Width = 57
Height = 33
Titight 00

TMiniForm
Left = 187
Top = 113
Width = 192
Height = 136
Hint = 'Please Resize Me !!!'
Caption = 'Mini-Tee-Charts. Enjoy
!!!'
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -14
Font.Name = 'System'
Font.Style = []
OldCreateOrder = True
ShowHint = True
OnCreate = FormCreate
OnResize = FormResize
PixelsPerInch = 96
TextHeight = 16
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 97
Height = 102
Align = alLeft
TabOrder = 0
<> object Chart1: TChart
Left = 1
Top = 57
Width = 95
Height = 44
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Pen.Visible = False
Foot.Font.Charset =
DEFAULT_CHARSET
Foot.Font.Color = clRed
Foot.Font.Height = -12
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsItalic]
Foot.Frame.Color = clScrollBar
MarginBottom = 0
MarginLeft = 0
MarginRight = 0
MarginTop = 0
Title.Frame.Color = clScrollBar

Font.Charset =
DEFAULT_CHARSET
Font.Color = clNavy
Font.Height = -16
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 3
OnClick = BitBtn14Click
Glyph.Data = {
end
<> object Panel2: TPanel
Left = 0
Top = 0
Width = 553
Height = 17
Caption = 'Database Charts'
Color = clYellow
Font.Charset =
DEFAULT CHARSET
_
Font.Color = clNavy
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 4
end
end
<> object BitBtn1: TBitBtn
Left = 472
Top = 376
Width = 97
Height = 33
Caption = ' &Close'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 2
Kind = bkClose
end
End
30
<> object MiniForm:

XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 58
Top = 75
end
end
<> object Chart2: TChart
Left = 1
Top = 1
Width = 95
Height = 56
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Pen.Visible = False
Foot.Font.Charset =
DEFAULT CHARSET
Foot.Font.Color = clRed
Foot.Font.Height = -12
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsItalic]
Foot.Frame.Color = clScrollBar
MarginBottom = 0
MarginLeft = 0
MarginRight = 0
MarginTop = 0
Title.Frame.Color = clScrollBar
Title.Text.Strings = (
'TChart Component')
Title.Visible = False
BottomAxis.Visible = False
Chart3DPercent = 25
Frame.Visible = False
LeftAxis.Visible = False
Legend.Visible = False
RightAxis.Visible = False
TopAxis.Visible = False
View3D = False
View3DWalls = False
Align = alTop
TabOrder = 1
<pre>&lt;&gt; object BarSeries1:</pre>
TBarSeries
ColorEachPoint = True
Marks.ArrowLength = 20

Title.Text.Strings = (
'TChart Component')
Title.Visible = False
BottomAxis.Visible = False
Chart3DPercent = 30
Frame.Visible = False
LeftAxis.Visible = False
Legend.Visible = False
RightAxis.Visible = False
TopAxis.Visible = False
View3DWalls = False
Align = alClient
TabOrder = 0
<> object LineSeries1:
TLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clRed
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 15
Top = 74
end
<> object LineSeries2:
TLineSeries
Marks.ArrowLength = 8
Marks. Visible = False
SeriesColor = clGreen
AfterDrawValues =
LineSeries2AfterDrawValues
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000

Title.Text.Strings = (
'TChart Component')
Title.Visible = False
AxisVisible = False
BottomAxis.Visible = False
ClipPoints = False
Frame.Visible = False
LeftAxis.Visible = False
Legend.Visible = False
RightAxis.Visible = False
TopAxis.Visible = False
View3DOptions.Elevation = 315
View3DOptions.Orthogonal =
False
View3DOptions.Perspective = 0
View3DOptions.Rotation = 360
View3DWalls = False
Align = alTop
TabOrder = 0
<> object PieSeries1:
TPieSeries
Marks.ArrowLength = 8
Marks.Style = smsLabelPercent
Marks. Visible = False
SeriesColor = clRed
Circled = True
OtherSlice.Text = 'Other'
PieValues.DateTime = False
PieValues.Name = 'Y'
PieValues.Multiplier =
1.00000000000000000
PieValues.Order = loNone
Left = 134
Top = 13
end
end
<> object Chart4: TChart
Left = 1
Top = 57
Width = 85
Height = 44
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Pen.Visible = False
Foot.Font.Charset =
DEFAULT CHARSET
Foot.Font.Color = clRed
Foot.Font.Height = -12
1 JULI UILLICIGIII — -12

N. C. X71 01 1 15 1
Marks.Visible = False
SeriesColor = clRed
BarPen.Visible = False
Dark3D = False
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 34
Top = 13
end
end
end
<> object Panel2: TPanel
Left = 97
Top = 0
Width = 87
Height = 102 Align = alClient
TabOrder = 1
<> object Chart3: TChart
Left = 1
Top = 1
Width = 85
Height = 56
AllowPanning = pmNone
AllowZoom = False
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Pen.Visible = False
Foot.Font.Charset =
DEFAULT_CHARSET
Foot.Font.Color = clRed
Foot.Font.Height = -12
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsItalic]
Foot.Frame.Color = clScrollBar
MarginBottom = 0
MarginLeft = 0
MarginRight = 0
MarginTop = 0
Title.Frame.Color = clScrollBar

SeriesColor = clGreen
AfterDrawValues =
LineSeries3AfterDrawValues
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.00000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 164
Top = 79
end
end
end
<> object Timer1: TTimer
Interval = 1
OnTimer = Timer1Timer
Left = 84
Top = 43
end
End
\\\\\\\\\
31
<> object LinkedTablesForm:
TLinkedTablesForm
Left = 185
Top = 113
Width = 604
Height = 457
Caption = 'TeeChart Linked Tables
Demo'
Color = clBtnFace
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter

Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsItalic]
Foot.Frame.Color = clScrollBar
MarginBottom = 0
MarginLeft = 0
MarginRight = 0
MarginTop = 0
Title.Frame.Color = clScrollBar
Title.Text.Strings = (
'TChart Component')
Title.Visible = False
BottomAxis.Visible = False
Frame.Visible = False
LeftAxis.Visible = False
Legend.Visible = False
RightAxis.Visible = False
TopAxis.Visible = False
View3D = False
View3DWalls = False
Align = alClient
TabOrder = 1
<> object AreaSeries1: TAreaSeries
Marks.ArrowLength = 8 Marks.Visible = False
SeriesColor = clRed AreaLinesPen.Visible = False
DrawArea = True
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 111
Top = 77
end
<> object LineSeries3:
TLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
•

Legend.Font.Color = dYellow   Legend.Font.Name = 'Arial'   Legend.Font.Style = [fsBold, fsItalic]   Legend.Frame.Color = clWhite   Legend.Frame.Width = 4   Legend.TextStyle = ltsRightValue   Legend.TopPos = 5   RightAxis.LabelsFont.Charset = DEFAULT CHARSET   BackWall.Brush.Style = bsClear   BackWall.Brush.Style = clear   BackWall.Brush.Style = bsClear   BackWall.Brush.Color = clWhite   BackWall.Brush.Style = bsClear   Back		
Legend.Font.Name = 'Arial'   Legend.Font.Style =  fsBold,	Legend.Font.Color = clYellow	PixelsPerInch = 96
Legend.Font.Style =  fsBold, fsItalic    Legend.Frame.Color = clWhite   Legend.Frame.Width = 4   Legend.TextStyle = ItsRightValue   Legend.TextStyle = ItsRightValue   Legend.TopPos = 5   RightAxis.LabelsFont.Charset = DEFAULT_CHARSET   RightAxis.LabelsFont.Color = clRed   RightAxis.LabelsFont.Name = 'Arial'   RightAxis.LabelsFont.Name = 'Arial'   Foot.Font.Color = clBlue   Foot.Font.Color = clBroot.Font.Name = 'Arial'   Foot.Font.Color = clScrollBar   Foot.Font.Color = clScrollBar   Title.Font.Color = clScrollBar   Title.Font.Color = clScrollBar   Title.Font.Style = [fsBold, fsItalic]   Title.Font.Style = [fsBold]	Legend.Font.Height = -16	TextHeight = 13
Legend.Frame.Color = clWhite	Legend.Font.Name = 'Arial'	<> object DBChart1:
Legend.Frame.Color = clWhite Legend.Trame.Width = 4 Legend.TopPos = 5 RightAxis.LabelsFont.Charset = DEFAULT CHARSET RightAxis.LabelsFont.Color = clRed RightAxis.LabelsFont.Name = 'Arial' RightAxis.LabelsFont.Name = 'Arial' RightAxis.TickLength = 12 RightAxis.Ticks.Width = 2 Align = alClient TabOrder = 0 <> object AreaSeries1: TaraSeries ColorEachPoint = True Marks.ArrowLength = 8 Marks.Visible = False DataSource = Table2 SeriesColor = clRed VertAxis = aRightAxis XLabelsSource = 'SaleDate' DrawArea = True Pointer.Style = psRectangle Pointer.Visible = False XValues.Name = 'X' XValues.Name = 'X' XValues.Name = 'X' XValues.Name in Color = clRed Yvalues.Name in Color = clRed RackWall.Brush.Color = clWhite BackWall.Pen.Color = clRed BackWall.Pen.Color = clBule Foot.Font.Color = clBlue Foot.Font.Color = clBule Foot.Font.Name in Color = clRed Foot.Font.Name in Color = clRed Title.Font.Style = [fsBold] Foot.Font.Color = clScrollBar Foot.Visible = False Title.Font.Name in Color = clRed Title.Font.Style in Galaria Title.Font.Style in	Legend.Font.Style = [fsBold,	<b>TDBChart</b>
Legend.Frame.Width = 4     Legend.TextStyle = ltsRightValue     Legend.TextStyle = ltsRightValue     Legend.TopPos = 5     RightAxis.LabelsFont.Charset =     DEFAULT_CHARSET     RightAxis.LabelsFont.Color = clRed     RightAxis.LabelsFont.Name =     Arial'     RightAxis.LabelsFont.Style =     fishtalic     RightAxis.Ticks.LabelsFont.Style =     fishtalic     RightAxis.Ticks.Color = clWhite     RightAxis.Ticks.Color = clWhite     RightAxis.Ticks.Color = clWhite     RightAxis.Ticks.Color = clWhite     RightAxis.Ticks.Width = 2     Align = alClient     TabOrder = 0     Camera = O     Ca	fsItalic]	Left = 0
Legend.TextStyle = ItsRightValue   Legend.TopPos = 5   BackWall.Brush.Color = clWhite   BackWall.Brush.Style = bsClear   BackWall.Brush.Style = bsClear   BackWall.Brush.Style = bsClear   BackWall.Pen.Style = psDashDot   Foot.Alignment = taRightJustify   Foot.Font.Charset = DEFAULT CHARSET   BackWall.Pen.Style = psDashDot   Foot.Alignment = taRightJustify   Foot.Font.Charset = DEFAULT CHARSET   Foot.Font.Color = clBlue   Foot.Font.Charset = DEFAULT CHARSET   Foot.Font.Height = -15   Foot.Font.Style = [fsBtalic]   RightAxis.Ticks.Color = clWhite   Foot.Font.Charset = DEFAULT CHARSET   Title.Font.Charset = DEFAULT CHARSET   Title.Font.Color = clScrollBar   Title.Font.Height = -16   Title.Font.Height = -16   Title.Font.Style = [fsBold, fsItalic]   Title.Font.Style = [fsBold, fsItalic]   Title.Font.Style = [fsBold, fsItalic]   Title.Font.Style = [fsBold, fsItalic]   Title.Text.Strings = (	Legend.Frame.Color = clWhite	Top = 241
Legend.TopPos = 5   RightAxis.LabelsFont.Charset = DEFAULT CHARSET   BackWall.Brush.Style = bsClear   BackWall.Pen.Style = psDashDot   Foot.Alignment = taRightJustify   Foot.Font.Charset = DEFAULT CHARSET   Foot.Font.Color = clBlue   Foot.Font.Color = clBlue   Foot.Font.Color = clBlue   Foot.Font.Name = 'Arial'   Foot.Font.Style = [fsBtalie]   RightAxis.Ticks.DelsFont.Style = [fsBtalie]   RightAxis.Ticks.Color = clWhite   RightAxis.Ticks.Width = 2   Align = alClient   TabOrder = 0   ColorEachPoint = True   Marks.ArrowLength = 8   Marks.Visible = False   DataSource = Table2   SeriesColor = clRed   VertAxis = aRightAxis   XLabelsSource = 'SaleDate'   DrawArea = True   Pointer.Style = psRectangle   Pointer.Visible = False   XValues.DateTime = False   XValues.DateTime = False   YValues.DateTime = False   YValues.Order = loAscending   YValues.Order = loAscending   YValues.Order = loAscending   YValues.Order = loNone   YValues.Order = loNone   YValues.Order = loNone   YValues.ValueSource =   Send. Font.Charset = Legend.Color = clBlue   Legend.Color = clAqua   Legend.Color = clBlue   Legend.Color = clBlue   Legend.Color = clBlue   Legend.Color = clBlue   Legend.Color = clAqua   Legend.Color = clBlue   Legend.Color = clBlue   Legend.Color = clAqua   Legend.Color = clBlue   Legend.Color =	Legend.Frame.Width = 4	Width = 596
RightAxis.LabelsFont.Color = clRed   RightAxis.LabelsFont.Height = - 15   RightAxis.LabelsFont.Name = 'Arial'   RightAxis.LabelsFont.Style =   BackWall.Pen.Style = psDashDot   Foot.Alignment = taRightJustify   Foot.Font.Color = clBlue   Foot.Font.Style = [fsBold]   Foot.Font.Color = clScrollBar   Title.Font.Style = [fsBold]   Foot.Font.Color = clScrollBar   Title.Font.Color = clScrollBar   Title.Font.Style = [fsBold]   Foot.Font.Color = clScrollBar   Title.Font.Color = clScrollBar   Title.Font.Style = [fsBold]   Foot.Font.Color = clScrollBar   Title.Font.Color = clScrollBar   Title.Font.Style = [fsBold]   Title.Font.Style = [fsBold]   Foot.Font.Color = clScrollBar   Title.Font.Style = [fsBold]   Title.Font.Colo	Legend.TextStyle = ltsRightValue	Height = 157
BackWall.Pen.Color = clAqua	Legend.TopPos = 5	BackWall.Brush.Color = clWhite
RightAxis.LabelsFont.Height = - 15 RightAxis.LabelsFont.Name = 'Arial' RightAxis.LabelsFont.Style =   EFAULT CHARSET   Foot.Font.Clolor = clBlue   Foot.Font.Clolor = clBlue   Foot.Font.Clolor = clBlue   Foot.Font.Clolor = clBlue   Foot.Font.Name = 'Arial'   Foot.Font.Name = 'Arial'   Foot.Font.Name = 'Arial'   Foot.Font.Name = 'Arial'   Foot.Font.Style =   fsBold   Foot.Font.Name = 'Arial'   Foot.Font.Clolor = clBlue   Foot.Font.Name = 'Arial'   Foot.Font.Clolor = clBlue   Foot.Font.Name = 'Arial'   Foot.Font.Clarset =   DEFAULT CHARSET   Foot.Font.Name = 'Arial'   Foot.Font.Clarset =   Foot.Font.Name = 'Arial'   Foot.Font.Clolor = clScrollBar   Foot.Font.Name = 'Arial'   Foot.Font.Clolor = clScrollBar   Foot.Font.Name = 'Arial'   Title.Font.Color = clFuchsia   Title.Font.Name = 'Arial'   Title.Font.Style =  fsBold, fsItalic    Title.	RightAxis.LabelsFont.Charset =	BackWall.Brush.Style = bsClear
College	DEFAULT_CHARSET	BackWall.Pen.Color = clAqua
RightAxis.LabelsFont.Name =  'Arial' RightAxis.LabelsFont.Style =  [fisItalic] RightAxis.TickLength = 12 RightAxis.Ticks.Color = clWhite RightAxis.Ticks.Width = 2 Align = alClient TabOrder = 0 > object AreaSeries1: TAreaSeries ColorEachPoint = True Marks.ArrowLength = 8 Marks.Visible = False DataSource = Table2 SeriesColor = clRed VertAxis = aRightAxis XLabelsSource = 'SaleDate' DrawArca = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.000000000000000000000000000000000000	RightAxis.LabelsFont.Color =	BackWall.Pen.Style = psDashDot
DEFAULT_CHARSET   Foot.Font.Color = clBlue   Foot.Font.Leight = -15   Foot.Font.Style =   Foot.Font.Styl	clRed	Foot.Alignment = taRightJustify
RightAxis.LabelsFont.Name = 'Arial'  RightAxis.LabelsFont.Style = [fsItalic] RightAxis.TickLength = 12 RightAxis.Ticks.Color = clWhite RightAxis.Ticks.Width = 2 Align = alClient TabOrder = 0 > object AreaSeries1: TAreaSeries ColorEachPoint = True Marks.ArrowLength = 8 Marks.Visible = False DataSource = Table2 SeriesColor = clRed VertAxis = aRightAxis XLabelsSource = 'SaleDate' DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Style = psRectangle Pointer.Style = psRectangle Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.DateTime = False YValues.Multiplier = 1.000000000000000000000000000000000000	RightAxis.LabelsFont.Height = -	Foot.Font.Charset =
'Arial'	15	DEFAULT_CHARSET
RightAxis.LabelsFont.Style =   fsltalic  RightAxis.TickLength = 12 RightAxis.Ticks.Color = clWhite RightAxis.Ticks.Color = clScrollBar Foot.Font.Style = [fsBold] Title.Font.Color = clPuchsia Title.Font.Color = clPuchsia Title.Font.Name = 'Arial' Title.Font.Style = [fsBold, fsltalic] Title.Font.Style = [fsBold, fsltalic] Title.Font.Style = [fsBold, fsltalic] Title.Font.Style = [fsBold, fsltalic] Title.Font.Name = 'Arial' Title.Font.Charset = DEFAULT _CHARSET  BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET  BottomAxis.LabelsFont.Charset = 'Arial' BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Style = [fsBold] Foot.Font.Name = 'Arial' Bo		Foot.Font.Color = clBlue
fsItalic  RightAxis.TickLength = 12   RightAxis.Ticks.Color = clWhite   RightAxis.Ticks.Width = 2   Align = alClient   TabOrder = 0   Title.Font.Charset = DEFAULT_CHARSET   Title.Font.Name = 'Arial'   Title.Font.Style = [fsBold, fsItalic]   Title.Font.Style = [fsBold, fsItalic]   Title.Font.Name = 'Arial'   Title.Font.Style = [fsBold, fsItalic]   Title.Font.Styl	'Arial'	Foot.Font.Height = -15
RightAxis.TickLength = 12 RightAxis.Ticks.Color = clWhite RightAxis.Ticks.Width = 2 Align = alClient TabOrder = 0 <> object AreaSeries1: TAreaSeries ColorEachPoint = True Marks.ArrowLength = 8 Marks.Visible = False DataSource = Table2 SeriesColor = clRed VertAxis = aRightAxis XLabelsSource = 'SaleDate' DrawArea = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.DateTime = False YValues.Order = loAscending YValues.Order = loAscending YValues.Order = loNone TabOrder = loNone Title.Font.Color = clFuchsia Title.Font.Visible = False Title.Font.Visible = Fa	, ,	Foot.Font.Name = 'Arial'
RightAxis.Ticks.Color = clWhite RightAxis.Ticks.Width = 2 Align = alClient TabOrder = 0 <> object AreaSeries1: TAreaSeries  ColorEachPoint = True Marks.ArrowLength = 8 Marks.Visible = False DataSource = Table2 SeriesColor = clRed VertAxis = aRightAxis XLabelsSource = 'SaleDate' DrawArea = True Pointer.InflateMargins = True Pointer.Visible = False XValues.DateTime = False XValues.Nultiplier = 1.0000000000000000000 XValues.Order = loAscending YValues.Multiplier = 1.000000000000000000000000000000000000		Foot.Font.Style = [fsBold]
RightAxis.Ticks.Width = 2 Align = alClient  TabOrder = 0  <> object AreaSeries1:  TAreaSeries  ColorEachPoint = True  Marks.ArrowLength = 8  Marks.Visible = False  DataSource = Table2  SeriesColor = clRed  VertAxis = aRightAxis  XLabelsSource = 'SaleDate'  DrawArea = True  Pointer.InflateMargins = True  Pointer.Visible = False  XValues.DateTime = False  XValues.Multiplier = 1.00000000000000000000000000000000000		Foot.Frame.Color = clScrollBar
Align = alClient TabOrder = 0  <> object AreaSeries1: TAreaSeries  ColorEachPoint = True Marks.ArrowLength = 8 Marks.Visible = False DataSource = Table2 SeriesColor = clRed VertAxis = aRightAxis XLabelsSource = 'SaleDate' DrawArea = True Pointer.InflateMargins = True Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Name = 'X' XValues.Order = loAscending YValues.DateTime = False YValues.Multiplier = 1.000000000000000000000000000000000000		Foot.Visible = False
TabOrder = 0  <> object AreaSeries1: TAreaSeries  ColorEachPoint = True Marks.ArrowLength = 8 Marks.Visible = False DataSource = Table2 SeriesColor = clRed VertAxis = aRightAxis XLabelsSource = 'SaleDate' DrawArea = True Pointer.InflateMargins = True Pointer.Visible = False XValues.DateTime = False XValues.Name = 'X' XValues.Name = 'X' XValues.Order = loAscending YValues.DateTime = False YValues.Multiplier = 1.00000000000000000000000000000000000		Title.Font.Charset =
<pre>ColorEachPoint = True     Marks.ArrowLength = 8     Marks.Visible = False     DataSource = Table2     SeriesColor = clRed     VertAxis = aRightAxis     XLabelsSource = 'SaleDate'     DrawArea = True     Pointer.InflateMargins = True     Pointer.Visible = False     XValues.DateTime = False     XValues.Multiplier =     1.000000000000000000000000000000000</pre>	C	DEFAULT_CHARSET
TAreaSeries  ColorEachPoint = True  Marks.ArrowLength = 8  Marks.Visible = False  DataSource = Table2  SeriesColor = clRed  VertAxis = aRightAxis  XLabelsSource = 'SaleDate'  DrawArea = True  Pointer.InflateMargins = True  Pointer.Visible = False  XValues.DateTime = False  XValues.DateTime = False  YValues.DateTime = False  YValues.DateTime = False  YValues.Name = 'Y'  YValues.Multiplier =  1.00000000000000000000000000000000000	TabOrder = 0	Title.Font.Color = clFuchsia
ColorEachPoint = True  Marks.ArrowLength = 8  Marks.Visible = False  DataSource = Table2  SeriesColor = clRed  VertAxis = aRightAxis  XLabelsSource = 'SaleDate'  DrawArea = True  Pointer.InflateMargins = True  Pointer.Visible = False  XValues.DateTime = False  XValues.Multiplier =  1.00000000000000000000000000000000000		Title.Font.Height = -16
Marks.ArrowLength = 8  Marks.Visible = False  DataSource = Table2  SeriesColor = clRed  VertAxis = aRightAxis  XLabelsSource = 'SaleDate'  DrawArea = True  Pointer.InflateMargins = True  Pointer.Visible = False  XValues.DateTime = False  XValues.Name = 'X'  XValues.Multiplier = 1.00000000000000000000000000000000000		Title.Font.Name = 'Arial'
Marks.Visible = False DataSource = Table2 SeriesColor = clRed VertAxis = aRightAxis XLabelsSource = 'SaleDate' DrawArea = True Pointer.InflateMargins = True Pointer.Visible = False XValues.DateTime = False XValues.Multiplier = 1.000000000000000000000000000000000000		Title.Font.Style = [fsBold, fsItalic]
DataSource = Table2 SeriesColor = clRed VertAxis = aRightAxis XLabelsSource = 'SaleDate' DrawArea = True Pointer.InflateMargins = True Pointer.Visible = False XValues.DateTime = False XValues.Multiplier = 1.000000000000000000 XValues.Order = loAscending YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000000000000000000000		Title.Frame.Color = clScrollBar
SeriesColor = clRed VertAxis = aRightAxis XLabelsSource = 'SaleDate' DrawArea = True Pointer.InflateMargins = True Pointer.Visible = False XValues.DateTime = False XValues.Multiplier = 1.000000000000000000000000000000000000		Title.Text.Strings = (
VertAxis = aRightAxis  XLabelsSource = 'SaleDate'  DrawArea = True  Pointer.InflateMargins = True  Pointer.Visible = psRectangle  Pointer.Visible = False  XValues.DateTime = False  XValues.Multiplier = 1.000000000000000000000000000000000000		'Linked Tables Chart Demo')
DEFAULT_CHARSET   BottomAxis.LabelsFont.Color = clGreen   BottomAxis.LabelsFont.Height = - 13   BottomAxis.LabelsFont.Name = 13   BottomAxis.LabelsFont.Name = 14   Yvalues.DateTime = False   Yvalues.Name = 'Y'   BottomAxis.LabelsFont.Name = 14   Yvalues.DateTime = False   Yvalues.DateTime = False   Tyvalues.DateTime = False   Yvalues.Name = 'Y'   BottomAxis.LabelsFont.Style = 15   BottomAxis.LabelsFont.Style = 16   SbottomAxis.LabelsFont.Style = 16   Sbott		BottomAxis.LabelsAngle = 90
DrawArea = True Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Multiplier = 1.000000000000000000000000000000000000		BottomAxis.LabelsFont.Charset =
Pointer.InflateMargins = True Pointer.Style = psRectangle Pointer.Visible = False  XValues.DateTime = False XValues.Multiplier = 1.000000000000000000000000000000000000		DEFAULT_CHARSET
Pointer.Style = psRectangle Pointer.Visible = False XValues.DateTime = False XValues.Multiplier = 1.000000000000000000 XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000 YValues.Multiplier = 1.0000000000000000000 YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000 YValues.Order = loNone YValues.Order = loNone YValues.Order = loNone YValues.ValueSource =  BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Style = [fsBold] BottomAxis.LabelsFon		BottomAxis.LabelsFont.Color =
Pointer.Visible = False  XValues.DateTime = False  XValues.Name = 'X'  XValues.Multiplier = 1.00000000000000000  XValues.Order = loAscending YValues.DateTime = False  YValues.Name = 'Y'  YValues.Multiplier = 1.00000000000000000  YValues.Multiplier = 1.000000000000000000  YValues.Order = loNone  YValues.ValueSource =  13  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Style = [fsBold]  BottomAxis.LabelsTyle = talText  Chart3DPercent = 35  Frame.Color = clAqua  Frame.Style = psDashDot  Legend.Color = clBlue  Legend.ColorWidth = 13  Legend.Font.Charset =		clGreen
XValues.DateTime = False XValues.Name = 'X' XValues.Multiplier = 1.00000000000000000 XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.000000000000000000 YValues.Multiplier = 1.000000000000000000000000000000000000		BottomAxis.LabelsFont.Height = -
XValues.Name = 'X'  XValues.Multiplier = 1.00000000000000000  XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y'  YValues.Multiplier = 1.00000000000000000  YValues.Order = loNone YValues.ValueSource =  'Arial'  BottomAxis.LabelsFont.Style = [fsBold]  BottomAxis.LabelStyle = talText  Chart3DPercent = 35  Frame.Color = clAqua  Frame.Style = psDashDot  Legend.Color = clBlue  Legend.ColorWidth = 13  Legend.Font.Charset =		
XValues.Multiplier = 1.00000000000000000000000000000000000	XValues.DateTime = False	BottomAxis.LabelsFont.Name =
1.000000000000000000000000000000000000	XValues.Name = 'X'	'Arial'
XValues.Order = loAscending YValues.DateTime = False YValues.Name = 'Y' YValues.Multiplier = 1.00000000000000000 YValues.Order = loNone YValues.ValueSource =  BottomAxis.LabelStyle = talText Chart3DPercent = 35 Frame.Color = clAqua Frame.Style = psDashDot Legend.Color = clBlue Legend.ColorWidth = 13 Legend.Font.Charset =	_	
YValues.DateTime = False YValues.Name = 'Y'  YValues.Multiplier = 1.00000000000000000 YValues.Order = loNone YValues.ValueSource =  Chart3DPercent = 35 Frame.Color = clAqua Frame.Style = psDashDot Legend.Color = clBlue Legend.ColorWidth = 13 Legend.Font.Charset =	1.000000000000000000	[fsBold]
YValues.Name = 'Y'  YValues.Multiplier =  1.00000000000000000000  YValues.Order = loNone  YValues.ValueSource =  Frame.Color = clAqua  Frame.Style = psDashDot  Legend.Color = clBlue  Legend.ColorWidth = 13  Legend.Font.Charset =	XValues.Order = loAscending	
YValues.Multiplier = Frame.Style = psDashDot 1.00000000000000000  YValues.Order = loNone YValues.ValueSource = Legend.ColorWidth = 13 Legend.Font.Charset =		
1.000000000000000000000000000000000000	YValues.Name = 'Y'	Frame.Color = clAqua
YValues.Order = loNone Legend.ColorWidth = 13 YValues.ValueSource = Legend.Font.Charset =	_	Frame.Style = psDashDot
YValues.ValueSource = Legend.Font.Charset =		Legend.Color = clBlue
		Legend.ColorWidth = 13
'AmountPaid' DEFAULT_CHARSET		S
	'AmountPaid'	DEFAULT_CHARSET

Left = 12
Top = 11
Width = 232
Height = 26
DataSource = DataSource1
VisibleButtons = [nbFirst,
nbPrior, nbNext, nbLast
TabOrder = 0
end
<> object BitBtn1: TBitBtn
Left = 286
Top = 8
Width = 95
Height = 33
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -15
Font.Name = 'Arial'
Font.Style = []
ParentFont = False
TabOrder = 1
Kind = bkClose
end
end
<> object DBGrid2: TDBGrid
Left = 0
Top = 129
Width = 596
Width = 596 Height = 112
Width = 596 Height = 112 Align = alTop
Width = 596 Height = 112 Align = alTop DataSource = DataSource2
Width = 596 Height = 112 Align = alTop DataSource = DataSource2 TabOrder = 4
Width = 596 Height = 112 Align = alTop DataSource = DataSource2 TabOrder = 4 TitleFont.Charset =
Width = 596 Height = 112 Align = alTop DataSource = DataSource2 TabOrder = 4 TitleFont.Charset = DEFAULT_CHARSET
Width = 596 Height = 112 Align = alTop DataSource = DataSource2 TabOrder = 4 TitleFont.Charset = DEFAULT_CHARSET TitleFont.Color = clBlack
Width = 596 Height = 112 Align = alTop DataSource = DataSource2 TabOrder = 4 TitleFont.Charset = DEFAULT_CHARSET TitleFont.Color = clBlack TitleFont.Height = -11
Width = 596 Height = 112 Align = alTop DataSource = DataSource2 TabOrder = 4 TitleFont.Charset = DEFAULT_CHARSET TitleFont.Color = clBlack TitleFont.Height = -11 TitleFont.Name = 'MS Sans Serif'
Width = 596 Height = 112 Align = alTop DataSource = DataSource2 TabOrder = 4 TitleFont.Charset = DEFAULT_CHARSET TitleFont.Color = clBlack TitleFont.Height = -11 TitleFont.Name = 'MS Sans Serif' TitleFont.Style = []
Width = 596 Height = 112 Align = alTop DataSource = DataSource2 TabOrder = 4 TitleFont.Charset = DEFAULT_CHARSET TitleFont.Color = clBlack TitleFont.Height = -11 TitleFont.Name = 'MS Sans Serif' TitleFont.Style = [] end
Width = 596 Height = 112 Align = alTop DataSource = DataSource2 TabOrder = 4 TitleFont.Charset = DEFAULT_CHARSET TitleFont.Color = clBlack TitleFont.Height = -11 TitleFont.Name = 'MS Sans Serif' TitleFont.Style = [] end <> object DataSource1:
Width = 596 Height = 112 Align = alTop DataSource = DataSource2 TabOrder = 4 TitleFont.Charset = DEFAULT_CHARSET TitleFont.Color = clBlack TitleFont.Height = -11 TitleFont.Name = 'MS Sans Serif' TitleFont.Style = [] end <> object DataSource1: TDataSource
Width = 596 Height = 112 Align = alTop DataSource = DataSource2 TabOrder = 4 TitleFont.Charset = DEFAULT_CHARSET TitleFont.Color = clBlack TitleFont.Height = -11 TitleFont.Name = 'MS Sans Serif' TitleFont.Style = [] end <> object DataSource1:
Width = 596 Height = 112 Align = alTop DataSource = DataSource2 TabOrder = 4 TitleFont.Charset = DEFAULT_CHARSET TitleFont.Color = clBlack TitleFont.Height = -11 TitleFont.Name = 'MS Sans Serif' TitleFont.Style = [] end <> object DataSource1: TDataSource
Width = 596 Height = 112 Align = alTop DataSource = DataSource2 TabOrder = 4 TitleFont.Charset = DEFAULT_CHARSET TitleFont.Color = clBlack TitleFont.Height = -11 TitleFont.Name = 'MS Sans Serif' TitleFont.Style = [] end <> object DataSource1: TDataSource DataSet = Table1
Width = 596 Height = 112 Align = alTop DataSource = DataSource2 TabOrder = 4 TitleFont.Charset = DEFAULT_CHARSET TitleFont.Color = clBlack TitleFont.Height = -11 TitleFont.Name = 'MS Sans Serif' TitleFont.Style = [] end <> object DataSource1: TDataSource DataSet = Table1 OnDataChange =
Width = 596 Height = 112 Align = alTop DataSource = DataSource2 TabOrder = 4 TitleFont.Charset = DEFAULT_CHARSET TitleFont.Color = clBlack TitleFont.Height = -11 TitleFont.Name = 'MS Sans Serif' TitleFont.Style = [] end <> object DataSource1: TDataSource DataSet = Table1 OnDataChange = DataSource1DataChange
Width = 596 Height = 112 Align = alTop DataSource = DataSource2 TabOrder = 4 TitleFont.Charset = DEFAULT_CHARSET TitleFont.Color = clBlack TitleFont.Height = -11 TitleFont.Name = 'MS Sans Serif' TitleFont.Style = [] end <> object DataSource1: TDataSource DataSet = Table1 OnDataChange = DataSource1DataChange Left = 339

<u>1</u>
Left = 40
Top = 258
end
end
<> object DBGrid1: TDBGrid
Left = 0
Top = 48
Width = 596
Height = 81
Align = alTop
DataSource = DataSource1
Options = [dgTitles, dgIndicator,
dgColumnResize, dgColLines,
dgRowLines, dgTabs, dgRowSelect,
dgAlwaysShowSelection,
dgConfirmDelete, dgCancelOnExit]
TabOrder = 1
TitleFont.Charset =
DEFAULT CHARSET
TitleFont.Color = clBlack
TitleFont.Height = -11
TitleFont.Name = 'MS Sans Serif'
TitleFont.Style = []
end
<> object Panel1: TPanel
Left = 0
Top = 398
Width = 596
Height = 25
Align = alBottom
Alignment = taLeftJustify
Font.Charset =
DEFAULT_CHARSET
Font.Color = clRed
Font.Height = -13
Font.Name = 'Arial'
Font.Style = [fsItalic]
ParentFont = False
TabOrder = 2
end
<> object Panel2: TPanel
Left = 0
Top = 0
Width = 596
Height = 48
Align = alTop
TabOrder = 3
<> object DBNavigator1:
TDBNavigator

FieldName = 'CustNo'
Required = True
Visible = False
end
<> object Table2SaleDate:
TDateTimeField
FieldName = 'SaleDate'
end SaleBate
<> object
Table2AmountPaid:
TCurrencyField
FieldName = 'AmountPaid'
end
end
<> object DataSource2:
<b>TDataSource</b>
DataSet = Table2
Left = 336
Top = 150
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
32
<> object PieForm: TPieForm
Left = 178
Top = 104
Width = 642
Height = 421
Caption = 'TeeChart Pie Series
Demo'
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Chart1: TChart
Left = 137
Top = 0
Width = 497
Height = 387
AllowPanning = pmNone
AllowZoom = False
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
BackWall.Pen.Visible = False
Foot.Alignment = taLeftJustify
i outhing intent talking usury

<u>1</u> _
<> object Table1: TTable
Active = True
DatabaseName = 'DBDEMOS'
TableName = 'CUSTOMER.DB'
Left = 268
Top = 203
<> object Table1CustNo:
TFloatField
DisplayWidth = 12
FieldName = 'CustNo'
end
<> object Table1Company:
TStringField
DisplayWidth = 29
FieldName = 'Company'
Size = 30 end
<> object Table1City:
TStringField DisplayWidth = 19
DisplayWidth = 18
FieldName = 'City'
Size = 15
end
<> object Table1State:
TStringField
DisplayWidth = 7
FieldName = 'State'
end
<> object Table1Country:
TStringField
DisplayWidth = 14
FieldName = 'Country'
end
end
<> object Table2: TTable
Active = True
DatabaseName = 'DBDEMOS'
IndexFieldNames = 'CustNo'
MasterFields = 'CustNo'
MasterSource = DataSource1
TableName = 'ORDERS.DB'
Left = 268
Top = 151
<> object Table2OrderNo:
TFloatField
FieldName = 'OrderNo'
end
<> object Table2CustNo:
TFloatField

GlipPoints = False	10 Y. 30 3	E / E / Cl
Frame.Visible = False   LeftAxis.Grid.Color = clScrollBar   LeftAxis.LabelsFont.Charset = DEFAULT CHARSET   LeftAxis.LabelsFont.Color = clWhite   LeftAxis.LabelsFont.Name = 'Arial'   Foot.Font.Style = [fsltalic]   Foot.Frame.Color = clScrollBar   Foot.False   Foot.Font.Style = [fsltalic]   Foot.Frame.Color = clScrollBar   Foot.False   Foot.Font.Style = [fsltalic]   Foot.Frame.Color = clScrollBar   Foot.False   Foot.Font.Style = [fsltalic]   Foot.Frame.Color = clScrollBar   Foot.Frame.Color = clScrollBar   Foot.Frame.Color = clScrollBar   Foot.Frame.Color = clScrollBar   Foot.Font.Style = [fsot.Font.Style = [foot.Frame.Color = clScrollBar   Foot.Frame.Color = clScrollBar   Foot.Frame.Color = clScrollBar   Foot.Frame.Color = clScrollBar   Foot.Font.Style = [foot.Frame.Color = clScrollBar   Foot.Font.Style = [foot.Font.Style = ]]   Title.Font.Style = [foot.Font.Bight = -16	[fsItalic]	Foot.Font.Charset =
LeftAxis.LabelsFont.Charset = DEFAULT CHARSET  LeftAxis.LabelsFont.Height = -13 LeftAxis.LabelsFont.Mame = 'Arial' LeftAxis.LabelsFont.Height = -13 LeftAxis.LabelsFont.Style = [] LeftAxis.LabelsFont.Style = [] LeftAxis.Title.Caption = 'Stock Price' LeftAxis.Title.Font.Charset = DEFAULT CHARSET LeftAxis.Title.Font.Color = clNavy LeftAxis.Title.Font.Color = clNavy LeftAxis.Title.Font.Name = 'Arial' Title.Font.Style = [fsBold] RightAxis.Grid.Color = clScrollBar View3DOptions.Perspective = 0 View3DOptions.Perspective = 0 View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DO		
LeftAxis.LabelsFont.Charset = DEFAULT CHARSET   Foot.Font.Style =  fsItalite  Foot.Frame.Color = clStrollBar   Foot.Frame.Color = sk54016   Gradient.EndColor = 8454016   Title.Font.Color = 8454016   Title.Font.Color = 8454016   Title.Font.Color = 8454016   Ti		
DEFAULT CHARSET  LeftAxis.LabelsFont.Color = clWhite  LeftAxis.LabelsFont.Height = -13  LeftAxis.LabelsFont.Name = 'Arial'  LeftAxis.LabelsFont.Style =     LeftAxis.Title.Font.Charset = DEFAULT CHARSET  LeftAxis.Title.Font.Color = clNavy  LeftAxis.Title.Font.Style =     LeftAxis.Title.Font.Style =     LeftAxis.Title.Font.Style =     LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Style =     LeftAxis.Title.Font.Style =     LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Style =     IfsBold   RightAxis.Grid.Color = clScrollBar  View3DOptions.Clevation = 315  View3DOptions.Perspective = 0  View3DOptions.Perspective = 0  View3DOptions.Rotation = 360  View3DOptions.Rotation = 360  View3DOptions.Rotation = 360  View3DOptions.Perspective = 0  View3DWalls = False  Align = alClient  Color = clSilver  TabOrder = 0  OnMouseMove = Chart I MouseMove  ———— object PieSeries1:  TPieSeries  Cursor = 2020  Marks.ArrowLength = 8  Marks.Style = smsLabelPercent  Marks.Visible = True  SeriesColor = clRed  OnClick = PieSeries1Click  OtherSlice.Text = 'Other'  PieValues.Name = 'Y'  PieValues.Multiplier =  Foot.Text.Strings = (  "Poot.Text.Strings = (  "Gradient.Direction = gdBottomTop  Gradient.StartColor = elSe5vollEar  Title.AdjustFrame = False  Title.AdjustFrame = False  Title.Font.Color = clBlue  Title.Font.Color = clBlue  Title.Font.Style =  fsBold, fsItalic   Title.Font.Name = 'Arial'  Title.Font.Style =  fsBold, fsItalic   Title.Font.Style =  fsBold, fsItalic   Title.Font.Style =  fsBold, fsItalic   Title.Font.Style =  fsBold, fsItalic   Title.Font.Arsert  DEFAULT CHARSET  Title.Font.Color = clScrollBar  Title.Font.Color = clScrollBar  Title.Font.Arsert = DEFAULT CHARSET  BottomAxis.LabelsFont.Color = clScrollBar  BottomAxis.LabelsFont.Color = clScrollBar  BottomAxis.LabelsFont.Style =     BottomAxis.LabelsFont.Style =     BottomAxis.LabelsFont.Style =     BottomAxis.Title.Font.Color = clTeal  BottomAxis.Title.Font.Color = clScrollBar  BottomAxis.Title.Font.Color = clScrollBa		
LeftAxis.LabelsFont.Color = clWhite	LeftAxis.LabelsFont.Charset =	Foot.Font.Name = 'Arial'
clWhite     LeftAxis.LabelsFont.Height = -13     LeftAxis.LabelsFont.Name = 'Arial'  LeftAxis.Title.Caption = 'Stock Price' LeftAxis.Title.Font.Charset = DEFAULT CHARSET LeftAxis.Title.Font.Color = clNavy LeftAxis.Title.Font.Color = clNavy LeftAxis.Title.Font.Style =    LeftAxis.Title.Font.Name = 'Arial' LeftAxis.Title.Font.Style =    fsBold  RightAxis.Grid.Color = clScrollBar TopAxis.Grid.Color = clScrollBar View3DOptions.Elevation = 315 View3DOptions.Perspective = 0 View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient Color = clSilver TabOrder = 0 OnMouseMove = Chart1MouseMove Chart2MouseMouse Chart1MouseMove Chart2MouseMouse Chart1MouseMove Chart2MouseMouse Chart1MouseMove Chart2MouseMouse Chart1MouseMove Chart1MouseMove Chart2MouseMouse Chart1MouseMove Chart2MouseMouse Chart2MouseMouse Chart2MouseMouse Chart2MouseMouse Chart1MouseMove Chart2MouseMouse Chart1MouseMove Chart2MouseMouse Chart2MouseMouse Chart1MouseMove Chart2MouseMouse Chart2MouseMouse Chart2Mouse	DEFAULT_CHARSET	
LeftAxis.LabelsFont.Name = 'Arial'  LeftAxis.LabelsFont.Style =    LeftAxis.Title.Caption = 'Stock Price'  LeftAxis.Title.Font.Charset = DEFAULT CHARSET  LeftAxis.Title.Font.Color = clNavy  LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Style = [fsBold]  RightAxis.Grid.Color = clScrollBar  TopAxis.Grid.Color = clScrollBar  View3DOptions.Devation = 315  View3DOptions.Perspective = 0 View3DOptions.Rotation = 360  View3DOptions.Rotation = 360  View3DOptions.Rotation = 360  View3DOptions.Rotation = 360  Color = clSilver  TabOrder = 0  OnMouseMove = Chart1MouseMove  Chart1MouseMove = DEFAULT CHARSET  BottomAxis.LabelsFont.Color = ClScrollBar  Botto	LeftAxis.LabelsFont.Color =	Foot.Frame.Color = clScrollBar
LeftAxis.LabelsFont.Name = 'Arial'   Gradient.Direction = gdBottomTop	clWhite	Foot.Text.Strings = (
Arial'	LeftAxis.LabelsFont.Height = -13	'')
LeftAxis.Title.Caption = 'Stock Price'  LeftAxis.Title.Font.Charset = DEFAULT_CHARSET  LeftAxis.Title.Font.Color = clNavy  LeftAxis.Title.Font.Height = -15  LeftAxis.Title.Font.Style =  fsBold  RightAxis.Grid.Color = clScrollBar View3DOptions.Color = clScrollBar View3DOptions.Perspective = 0 View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DOptions.Rotation = 360 View3DOptions.Rotation = 360 View3DOptions.Perspective = 0 OnMouseMove = Clart1MouseMove  Cloor = clSilver  TabOrder = 0 OnMouseMove = Chart1MouseMove  Clors = 2020 Marks.ArrowLength = 8 Marks.Style = smsLabelPercent Marks.Visible = True  SeriesColor = clRed OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.DateTime = False PieValues.Multiplier =  Gradient.StartColor = 8454016 Gradient.StartColor = 1tle.Adjustframe = False  DEFAULT CHARSET  Title.Font.Name = 'Arial' Title.Font.Color = clScrollBar Title.Font.Name = 'Arial' Titl	LeftAxis.LabelsFont.Name =	Gradient.Direction =
LeftAxis.Title.Caption = 'Stock Price'  LeftAxis.Title.Font.Charset = DEFAULT CHARSET  LeftAxis.Title.Font.Color = clNavy  LeftAxis.Title.Font.Height = -15  LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Style = [fsBold]  RightAxis.Grid.Color = clScrollBar  TopAxis.Grid.Color = clScrollBar  View3DOptions.Elevation = 315  View3DOptions.Perspective = 0 View3DOptions.Perspective = 0 View3DOptions.Rotation = 360  BottomAxis.LabelsFont.Color = clScrollBar  BottomAxis.LabelsFont.Color = clScrollBar  BottomAxis.LabelsFont.Charset = DEFAULT CHARSET  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.Title.Font.Name = 'Arial'  BottomAxis.Title.Font.Color = clScrollBar  BottomAxis.Title.Font.Color = clScrollBar  Title.Font.Color = clBrue  Title.Font.Color = clBrue  Title.Font.Color = clBrue  Title.Font.Name = 'Arial'  Title.Font.Name = 'Arial'  BottomAxis.LabelsFont.Color = clScrollBar  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.Title.Font.Color = clScrollBar	'Arial'	gdBottomTop
LeftAxis.Title.Caption = 'Stock Price'  LeftAxis.Title.Font.Charset = DEFAULT CHARSET  LeftAxis.Title.Font.Color = clNavy  LeftAxis.Title.Font.Height = -15  LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Style = [fsBold]  RightAxis.Grid.Color = clScrollBar  TopAxis.Grid.Color = clScrollBar  View3DOptions.Elevation = 315  View3DOptions.Perspective = 0 View3DOptions.Perspective = 0 View3DOptions.Rotation = 360  BottomAxis.LabelsFont.Color = clScrollBar  BottomAxis.LabelsFont.Color = clScrollBar  BottomAxis.LabelsFont.Charset = DEFAULT CHARSET  BottomAxis.LabelsFont.Color = clScrollBar  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.Title.Font.Color = clScrollBar  BottomAxis.LabelsFont.Name = 'Ari	LeftAxis.LabelsFont.Style = []	Gradient.EndColor = 8454016
Price'   Gradient.Visible = True		Gradient.StartColor = 8454143
DEFAULT CHARSET  LeftAxis.Title.Font.Color = clNavy  LeftAxis.Title.Font.Height = -15  LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Style = [fsBold]  RightAxis.Grid.Color = clScrollBar  TopAxis.Grid.Color = clScrollBar  View3DOptions.Elevation = 315  View3DOptions.Orthogonal = False  View3DOptions.Perspective = 0  View3DOptions.Rotation = 360  View3DOptions.Rotation = 360  View3DOptions.Rotation = 360  View3DOptions.Rotation = 360  View3DOptions.Perspective = 0  OnMouseMove = Chart1MouseMove  Chart1MouseMove = Chart1MouseMove  Cursor = 2020  Marks.ArrowLength = 8  Marks.Style = smsLabelPercent Marks.Visible = True  SeriesColor = clRed OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.DateTime = False PieValues.Multiplier =  Title.Font.Color = clBlue Title.Font.Color = clBrue Title.Font.Color = clBrue Title.Font.Color = clBrue Title.Font.Color = clBrue Title.Font.Name = 'Arial' Title.Font.Color = clBrue Title.Font.Name = 'Arial' Title.Font.Color = clBrue Title.Font.Colo	_	Gradient.Visible = True
DEFAULT CHARSET  LeftAxis.Title.Font.Color = clNavy  LeftAxis.Title.Font.Height = -15  LeftAxis.Title.Font.Name = 'Arial'  LeftAxis.Title.Font.Style = [fsBold]  RightAxis.Grid.Color = clScrollBar  TopAxis.Grid.Color = clScrollBar  View3DOptions.Drthogonal = False  View3DOptions.Orthogonal = False  View3DOptions.Perspective = 0  View3DOptions.Rotation = 360  View3DOptions.Rotation = 360  View3DOptions.Rotation = 360  View3DOptions.Rotation = 360  View3DOptions.Perspective = 0  OnMouseMove = Chart1MouseMove  Chart1MouseMove = Chart1MouseMove  Cursor = 2020  Marks.ArrowLength = 8  Marks.Style = smsLabelPercent Marks.Visible = True  SeriesColor = clRed OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.DateTime = False PieValues.Multiplier =  Title.Font.Color = clBlue Title.Font.Height = -16 Title.Font.Name = 'Arial' Title.Font.Style = [fsBold, fsItalic} Title.Font.Style = fslomanis* Title.Font.Style = [fsBold, fsItalic} Title.Font.Style = [fsBold, fsItalic} Title.Font.Style = [fsBold, fslaic} Title.Font.Style = [fsBold, fslaic} Title.Font.Style = [fsBold, fslaic} Title.Fo	LeftAxis.Title.Font.Charset =	Title.AdjustFrame = False
LeftAxis.Title.Font.Color = clNavy  LeftAxis.Title.Font.Height = -15 LeftAxis.Title.Font.Name = 'Arial' LeftAxis.Title.Font.Style = [fsBold] RightAxis.Grid.Color = clScrollBar TopAxis.Grid.Color = clScrollBar View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DOptions.Rotation = 360 View3DOptions.Rotation = 360 View3DOptions.Rotation = 360 View3DOptions.Perspective = 0 OnMouseMove = Chart1MouseMove Chart1MouseMove Chart1MouseMove Chart1MouseMove Cursor = 2020 Marks.ArrowLength = 8 Marks.Style = smsLabelPercent Marks.Visible = True SeriesColor = clRed OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.DateTime = False PieValues.Multiplier =  Title.Font.Color = clBlue Title.Font.Name = 'Arial' Title.Font.Name = 'Arial' Title.Font.Name = 'Arial' Title.Font.Color = clBruc Title.Font.Name = 'Arial' Title.Font.Name = 'Arial' Title.Font.Color = clBruc Title.Font.Name = 'Arial' Title.Font.Name = 'Arial' Title.Font.Name = 'Arial' Title.Font.Name = 'Arial' Title.Font.Color = clBruc Title.Font.Name = 'Arial' Title.Font.Color = clBruc Title.Font.Name = 'Arial' Title.	DEFAULT_CHARSET	·
ClNavy	LeftAxis.Title.Font.Color =	
LeftAxis.Title.Font.Name = 'Arial' LeftAxis.Title.Font.Style =  [fsBold] RightAxis.Grid.Color =  clScrollBar TopAxis.Grid.Color = clScrollBar View3DOptions.Elevation = 315 View3DOptions.Orthogonal =  False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DOptions.Perspective = 0 OnMouseMove =  Chart MouseMove  <> object PieSeries1: TPieSeries Cursor = 2020 Marks.ArrowLength = 8 Marks.Style = smsLabelPercent Marks.Visible = True SeriesColor = clRed OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.Name = 'Y' PieValues.Multiplier =  Title.Font.Color = clBlue Title.Font.Name = 'Arial' Title.Font.Style = [fsBold, fsItalic] Title.Font.Style = [fsBold, fsItalic] Title.Font.Style = [fsBold, fsItalic] Title.Font.Name = 'Arial' Title.Font.Name = 'Arial	clNavy	
LeftAxis.Title.Font.Name = 'Arial' LeftAxis.Title.Font.Style =  fsBold  RightAxis.Grid.Color = clScrollBar clScrollBar TopAxis.Grid.Color = clScrollBar View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient Color = clSilver TabOrder = 0 OnMouseMove = Chart I MouseMove <> object PieSeries1: TPieSeries Cursor = 2020 Marks.ArrowLength = 8 Marks.Style = smsLabelPercent Marks.Visible = True SeriesColor = clRed OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.DateTime = False PieValues.Multiplier =  Title.Font.Height = -16 Title.Font.Name = 'Arial' Title.Font.Style = [fsBold, fsItalic] Title.Font.Style = [fsDold fill] Title.Font.Style = [fsBold, fsItalic] Title.Font.Style = [fsDold fill] Title.Font.Style = [fsDold fill] Title.Font.Style = [fsDold fill] Satisfactorial Title.Font.Color = clScrollBar Title.Text.Strings = (	LeftAxis.Title.Font.Height = -15	
LeftAxis.Title.Font.Style =   [fsBold]		
[fsBold]   RightAxis.Grid.Color = clScrollBar   Title.Font.Style = [fsBold, fsItalic]   Title.Frame.Color = clScrollBar   Title.Frame.Color = clScrollBar   Title.Frame.Color = clScrollBar   Title.Text.Strings = (	LeftAxis.Title.Font.Style =	
RightAxis.Grid.Color = clScrollBar  TopAxis.Grid.Color = clScrollBar  View3DOptions.Elevation = 315  View3DOptions.Perspective = 0  View3DOptions.Perspective = 0  View3DOptions.Rotation = 360  View3DOptions.Rotation = 360  View3DWalls = False  Align = alClient  Color = clSilver  TabOrder = 0  OnMouseMove =  Chart1MouseMove  <> object PieSeries1:  TPieSeries  Cursor = 2020  Marks.ArrowLength = 8  Marks.Style = smsLabelPercent  Marks.Visible = True  SeriesColor = clRed  OnClick = PieSeries1Click  OtherSlice.Text = 'Other'  PieValues.Name = 'Y'  PieValues.Multiplier =  Title.Frame.Color = clScrollBar  Title.Text.Strings = ( 'Pie Chart Series Demo.')  AxisVisible = False  BottomAxis.CabelsAngle = 90  BottomAxis.LabelsFont.Charset = DEFAULT CHARSET  BottomAxis.LabelsFont.Height = -12  BottomAxis.LabelsFont.Style =     BottomAxis.Title.Font.Color = clScrollBar  Title.Fext.Strings = ( 'Pie Chart Series Demo.')  AxisVisible = False  BottomAxis.LabelsAngle = 90  BottomAxis.LabelsFont.Charset = DEFAULT CHARSET  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.Title.Font.Charset = DEFAULT CHARSET  BottomAxis.Title.Font.Color = clScrollBar  Title.Text.Strings = ( 'Pie Chart Series Demo.')  AxisVisible = False  BottomAxis.LabelsAngle = 90  BottomAxis.LabelsFont.Charset = DEFAULT CHARSET  BottomAxis.LabelsFont.Style =    BottomAxis.Title.Font.Charset = DEFAULT CHARSET  BottomAxis.Title.Font.Color = clScrollBar  Title.Text.Strings = ( 'Pie Chart Series Demo.')  AxisVisible = False  BottomAxis.LabelsFont.Charset = DEFAULT CHARSET  BottomAxis.LabelsFont.Style =    BottomAxis.Title.Font.Charset = DEFAULT CHARSET  BottomAxis.Title.Font.Charset = DEFAULT CHARSET	-	
clScrollBar  TopAxis.Grid.Color = clScrollBar View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient Color = clSilver TabOrder = 0 OnMouseMove = Chart1MouseMove <> object PieSeries1: TPieSeries Cursor = 2020 Marks.ArrowLength = 8 Marks.Style = smsLabelPercent Marks.Visible = True SeriesColor = clRed OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.Name = 'Y' PieValues.Name = 'Y' PieValues.Multiplier =  Title.Text.Strings = ( 'Pie Chart Series Demo.') AxisVisible = False BottomAxis.CabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT CHARSET BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Name = 'Arial'	RightAxis.Grid.Color =	
TopAxis.Grid.Color = clScrollBar View3DOptions.Devation = 315 View3DOptions.Orthogonal = False  View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient Color = clSilver TabOrder = 0 OnMouseMove = Chart1MouseMove <> object PieSeries1: TPieSeries  Cursor = 2020 Marks.ArrowLength = 8 Marks.Style = smsLabelPercent Marks.Visible = True SeriesColor = clRed OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.Name = 'Y' PieValues.Multiplier =  'Pie Chart Series Demo.') AxisVisible = False BottomAxis.Grid.Color = clScrollBar BottomAxis.LabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT CHARSET BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Name = 'Arial'	clScrollBar	
View3DOptions.Elevation = 315 View3DOptions.Orthogonal = False View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient Color = clSilver TabOrder = 0 OnMouseMove = Chart1MouseMove <> object PieSeries1: TPieSeries Cursor = 2020 Marks.ArrowLength = 8 Marks.Style = smsLabelPercent Marks.Visible = True SeriesColor = clRed OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.Name = 'Y' PieValues.Multiplier =  AxisVisible = False BottomAxis.Color = clScrollBar BottomAxis.LabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT CHARSET BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Font.Charset = DEFAULT CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Name = 'Arial'	TopAxis.Grid.Color = clScrollBar	
View3DOptions.Orthogonal = False  View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient Color = clSilver TabOrder = 0 OnMouseMove = Chart1MouseMove <> object PieSeries1: TPieSeries Cursor = 2020 Marks.ArrowLength = 8 Marks.Style = smsLabelPercent Marks.Visible = True SeriesColor = clRed OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.DateTime = False PieValues.Multiplier =  BottomAxis.Grid.Color = clScrollBar BottomAxis.LabelsAngle = 90 BottomAxis.LabelsFont.Charset = DEFAULT CHARSET BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Font.Charset = DEFAULT CHARSET BottomAxis.LabelsFont.Name = 'Arial'		,
False  View3DOptions.Perspective = 0  View3DOptions.Rotation = 360  View3DWalls = False  Align = alClient  Color = clSilver  TabOrder = 0  OnMouseMove = Chart1MouseMove  <> object PieSeries1:  TPieSeries  Cursor = 2020  Marks.ArrowLength = 8  Marks.Visible = True  SeriesColor = clRed  OnClick = PieSeries1Click  OtherSlice.Text = 'Other'  PieValues.Name = 'Y'  PieValues.Multiplier =  ClScrollBar  BottomAxis.LabelsAngle = 90  BottomAxis.LabelsFont.Charset =  DEFAULT CHARSET  BottomAxis.LabelsFont.Height = -  12  BottomAxis.LabelsFont.Name =  'Arial'  BottomAxis.LabelsFont.Color = clTeal  BottomAxis.LabelsFont.Name =  'Arial'  BottomAxis.Title.Font.Charset = DEFAULT CHARSET  BottomAxis.Title.Font.Charset = DEFAULT CHARSET  BottomAxis.Title.Font.Height = -  16  BottomAxis.Title.Font.Name = 'Arial'		
View3DOptions.Perspective = 0 View3DOptions.Rotation = 360 View3DWalls = False Align = alClient Color = clSilver TabOrder = 0 OnMouseMove = Chart1MouseMove <> object PieSeries1: TPieSeries Cursor = 2020 Marks.ArrowLength = 8 Marks.Visible = True SeriesColor = clRed OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.Name = 'Y' PieValues.Multiplier =  BottomAxis.LabelsFont.Color = clTeal BottomAxis.LabelsFont.Height = -12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Name = 'Arial'	False	
View3DOptions.Rotation = 360 View3DWalls = False Align = alClient Color = clSilver TabOrder = 0 OnMouseMove = Chart1MouseMove <> object PieSeries1: TPieSeries Cursor = 2020 Marks.ArrowLength = 8 Marks.Style = smsLabelPercent Marks.Visible = True SeriesColor = clRed OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.DateTime = False PieValues.Multiplier =  BottomAxis.LabelsFont.Color = clTeal BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Caption = 'Stock Market Date' BottomAxis.Title.Font.Charset = DEFAULT CHARSET BottomAxis.Title.Font.Style = [] BottomAxis.LabelsFont.Charset = 'Arial' BottomAxis.LabelsFont.Charset = 'Arial' BottomAxis.LabelsFont.Name = 'Arial'	View3DOptions.Perspective = 0	
View3DWalls = False  Align = alClient Color = clSilver TabOrder = 0 OnMouseMove = Chart1MouseMove <> object PieSeries1: TPieSeries Cursor = 2020 Marks.ArrowLength = 8 Marks.Style = smsLabelPercent Marks.Visible = True SeriesColor = clRed OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.DateTime = False PieValues.Multiplier =  DEFAULT CHARSET BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Caption = 'Stock Market Date' BottomAxis.Title.Font.Charset = DEFAULT CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Name = 'Arial'	View3DOptions.Rotation = 360	
Align = alClient Color = clSilver TabOrder = 0 OnMouseMove = Chart1MouseMove <> object PieSeries1: TPieSeries Cursor = 2020 Marks.ArrowLength = 8 Marks.Style = smsLabelPercent Marks.Visible = True SeriesColor = clRed OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.DateTime = False PieValues.Multiplier =  BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Caption = 'Stock Market Date' BottomAxis.Title.Font.Charset = DEFAULT_CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.LabelsFont.Name = 'Arial'		
Color = clSilver TabOrder = 0 OnMouseMove = Chart1MouseMove <> object PieSeries1: TPieSeries Cursor = 2020 Marks.ArrowLength = 8 Marks.Style = smsLabelPercent Marks.Visible = True SeriesColor = clRed OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.DateTime = False PieValues.Multiplier =  clTeal BottomAxis.LabelsFont.Height = - 12 BottomAxis.LabelsFont.Name = 'Arial' BottomAxis.Title.Caption = 'Stock Market Date' BottomAxis.Title.Font.Charset = DEFAULT CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Name = 'Arial'	Align = alClient	
TabOrder = 0 OnMouseMove = 12 Chart1MouseMove   SottomAxis.LabelsFont.Height = -12 TPieSeries   BottomAxis.LabelsFont.Name = 'Arial'   TPieSeries   BottomAxis.LabelsFont.Name = 'Arial'   TPieSeries   BottomAxis.LabelsFont.Name = 'Arial'   BottomAxis.LabelsFont.Name = 'Arial'   BottomAxis.LabelsFont.Name = 'Arial'   BottomAxis.Title.Caption = 'Stock   Market Date'   BottomAxis.Title.Font.Charset =   DEFAULT_CHARSET   BottomAxis.Tit		
OnMouseMove = Chart1MouseMove   BottomAxis.LabelsFont.Name = 'Arial'    TPieSeries   Cursor = 2020   BottomAxis.LabelsFont.Style = []    Marks.ArrowLength = 8   Marks.Style = smsLabelPercent   Marks.Visible = True    SeriesColor = clRed   OnClick = PieSeries1Click    OtherSlice.Text = 'Other'    PieValues.DateTime = False    PieValues.Multiplier =    12   BottomAxis.LabelsFont.Name = 'Arial'    BottomAxis.Title.Caption = 'Stock    Market Date'    BottomAxis.Title.Font.Charset =    DEFAULT CHARSET    BottomAxis.Title.Font.Charset =    CHARSET    BottomAxis.Title.Font.Charset =    DEFAULT CHARSET    BottomAxis.Title.Font.Charset    Bott	TabOrder = 0	
Chart1MouseMove  <> object PieSeries1:  TPieSeries  Cursor = 2020  Marks.ArrowLength = 8  Marks.Style = smsLabelPercent  Marks.Visible = True  SeriesColor = clRed  OnClick = PieSeries1Click  OtherSlice.Text = 'Other'  PieValues.DateTime = False  PieValues.Name = 'Y'  PieValues.Multiplier =  BottomAxis.LabelsFont.Name = 'Arial'  BottomAxis.LabelsFont.Style = []  BottomAxis.Title.Caption = 'Stock Market Date'  BottomAxis.Title.Font.Charset = DEFAULT CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Name = 'Arial'		
<pre> &lt;&gt; object PieSeries1:    TPieSeries    Cursor = 2020         Marks.ArrowLength = 8         Marks.Style = smsLabelPercent         Marks.Visible = True         SeriesColor = clRed         OnClick = PieSeries1Click         OtherSlice.Text = 'Other'         PieValues.DateTime = False         PieValues.Name = 'Y'         PieValues.Multiplier =</pre>		
TPieSeries Cursor = 2020 Marks.ArrowLength = 8 Marks.Style = smsLabelPercent Marks.Visible = True SeriesColor = clRed OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.DateTime = False PieValues.Multiplier =  BottomAxis.LabelsFont.Style = [] BottomAxis.Title.Caption = 'Stock Market Date' BottomAxis.Title.Font.Charset = DEFAULT CHARSET BottomAxis.Title.Font.Color = clGreen		
Cursor = 2020  Marks.ArrowLength = 8  Marks.Style = smsLabelPercent  Marks.Visible = True  SeriesColor = clRed  OnClick = PieSeries1Click  OtherSlice.Text = 'Other'  PieValues.DateTime = False  PieValues.Multiplier =  BottomAxis.Title.Caption = 'Stock Market Date'  BottomAxis.Title.Font.Charset = DEFAULT CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Height = -16	<u> </u>	
Marks.ArrowLength = 8  Marks.Style = smsLabelPercent  Marks.Visible = True  SeriesColor = clRed  OnClick = PieSeries1Click  OtherSlice.Text = 'Other'  PieValues.DateTime = False  PieValues.Name = 'Y'  PieValues.Multiplier =  Market Date'  BottomAxis.Title.Font.Charset = DEFAULT_CHARSET  BottomAxis.Title.Font.Color = clGreen  BottomAxis.Title.Font.Height = -16  BottomAxis.Title.Font.Name = 'Y'  Arial'	Cursor = 2020	v 11
Marks.Style = smsLabelPercent Marks.Visible = True SeriesColor = clRed OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.DateTime = False PieValues.Name = 'Y' PieValues.Multiplier =  BottomAxis.Title.Font.Charset = DEFAULT CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Name = 'Y' Yrial'	Marks.ArrowLength = 8	_
Marks.Visible = True SeriesColor = clRed OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.DateTime = False PieValues.Name = 'Y' PieValues.Multiplier =  DEFAULT CHARSET BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Name = 'Yrial'		
SeriesColor = clRed OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.DateTime = False PieValues.Name = 'Y' PieValues.Multiplier =  BottomAxis.Title.Font.Color = clGreen BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Height = - 16 BottomAxis.Title.Font.Height = - 16 Varial'	•	
OnClick = PieSeries1Click OtherSlice.Text = 'Other' PieValues.DateTime = False PieValues.Name = 'Y' PieValues.Multiplier = ClGreen BottomAxis.Title.Font.Height = -16 BottomAxis.Title.Font.Name = 'Arial'		_
OtherSlice.Text = 'Other' PieValues.DateTime = False PieValues.Name = 'Y' PieValues.Multiplier =  BottomAxis.Title.Font.Height = - 16  BottomAxis.Title.Font.Name = 'Arial'		
PieValues.DateTime = False PieValues.Name = 'Y' PieValues.Multiplier =   16  BottomAxis.Title.Font.Name = 'Arial'		
PieValues.Name = 'Y' PieValues.Multiplier = BottomAxis.Title.Font.Name = 'Arial'		
PieValues.Multiplier = 'Arial'		
	-	BottomAxis.Title.Font.Style =

Width = 109
Height = 101
Caption = 'Legend Position:'
ItemIndex = 1
Items.Strings = (
'Left'
'Right'
'Top'
'Bottom')
TabOrder = 5
OnClick = RadioGroup1Click
end
<> object RadioGroup2:
TRadioGroup
Left = 12
Top = 80
Width = 109
Height = 156
Caption = 'Efects:'
TabOrder = 8
end
<> object CheckBox2:
TCheckBox
Left = 20
Top = 104
Width = 41
Height = 17
Caption = '&3D'
Checked = True
State = cbChecked
TabOrder = 1
OnClick = CheckBox2Click
end
<> object CheckBox3:
TCheckBox
Left = 20
Top = 130
Width = 61
Height = 17
Caption = '&Circled'
TabOrder = 2
OnClick = CheckBox3Click
end
<> object CheckBox4:
TCheckBox
Left = 20
Top = 156
<b>Width = 97</b>

PieValues.Order = loNone
Left = 78
Top = 94
end
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 137
Height = 387
Align = alLeft
TabOrder = 1
<> object Shape1: TShape
Left = 14
Top = 40
Width = 107
Height = 33
HelpType = htKeyword
end
<> object CheckBox1:
TCheckBox
Left = 15
<b>Top</b> = 6
Width = 105
Height = 29
Caption = '&Animate !!!'
Font.Charset =
DEFAULT CHARSET
Font.Color = clBlack
Font.Height = -17
Font.Name = 'Arial'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 0
OnClick = CheckBox1Click
end
<> object BitBtn3: TBitBtn
Left = 20
Top = 352
Width = 89
Height = 29
TabOrder = 3
Kind = bkClose
end
<> object RadioGroup1:
TRadioGroup
Left = 12
Top = 244

# <u>AC.&P</u>

Scroll Example'
Color = clBtnFace
Font.Charset =
DEFAULT_CHARSET
Font.Color = clBlack
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Chart1: TChart
Left = 0
Top = 53
Width = 574
Height = 323
BackImageInside = True
BackImageMode = pbmTile
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
Foot.Font.Charset =
DEFAULT_CHARSET
Foot.Font.Color = clRed
Foot.Font.Height = -12
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsItalic]
Foot.Frame.Color = clScrollBar
Title.Frame.Color = clScrollBar
Title.Text.Strings = (
'TChart Component')
BottomAxis.Grid.Color =
clScrollBar
LeftAxis.Grid.Color = clScrollBar
RightAxis.Grid.Color =
clScrollBar
TopAxis.Grid.Color = clScrollBar
Align = alClient
TabOrder = 0
<> object LineSeries1:
TLineSeries
Marks.ArrowLength = 8
Marks. Visible = False
SeriesColor = clRed
OnAfterAdd =
LineSeries1AfterAdd
Pointer.InflateMargins = True

Height = 17
Caption = 'Use & Patterns'
TabOrder = 4
OnClick = CheckBox4Click
end
<> object CheckBox5:
TCheckBox
Left = 20
Top = 182
Width = 61
Height = 17
Caption = '&Marks'
Checked = True
State = cbChecked
TabOrder = 6
OnClick = CheckBox5Click
end
<> object CheckBox6:
TCheckBox
Left = 20
Top = 208
Width = 65
Height = 17
Caption = '&Gradient'
Checked = True
State = cbChecked
TabOrder = 7
OnClick = CheckBox6Click
end
end
<> object Timer1: TTimer
Enabled = False
Interval = 1
OnTimer = Timer1Timer
Left = 114
Top = 294
end
End
111111111111111111111111111111111111111
33
<> object ScrollForm:
TScrollForm
Left = 191
Top = 113
Width = 582
Height = 410
ActiveControl = Button1
Caption = 'TeeChart - Automatic

TabOrder = 2
OnClick = CBVerticalClick
end
end
End
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
34
<> object LegendForm: TLegendForm
Left = 186
Top = 115
Width = 543
Height = 385  Caption = 'TeeChart Customized
Legend Example'
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object PaintBox1:
TPaintBox
Left = 0
Top = 0
Width = 535
Height = 52
HelpType = htKeyword
Align = alTop
end
<> object Label1: TLabel
Left = 8
Top = 4
Width = 123
Height = 13
HelpType = htKeyword
Caption = 'The Chart Legend is
here:'
end
<> object Chart1: TChart
Left = 153
Top = 52
Width = 382
Height = 299
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
Foot.Frame.Color = clScrollBar

Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.0000000000000000000
YValues.Order = loNone
Left = 119
Top = 94
end
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 574
Height = 53
Align = alTop
TabOrder = 1
<> object Button1: TButton
Left = 191
Top = 9
Width = 250
Height = 33
Caption = 'Click here to add a
new point and scroll !!!'
TabOrder = 0
OnClick = Button1Click
end
<> object BitBtn3: TBitBtn
Left = 8
Top = 12
Width = 89
Height = 33
TabOrder = 1
Kind = bkClose
end
<> object CBVertical: TCheckBox
Left = 110
Top = 18
Width = 71
Height = 17
Caption = '&Vertical'

# <u>AC.&P</u>

end  <> object Panel1: TPanel  Left = 0  Top = 52  Width = 153  Height = 299  Align = alLeft  TabOrder = 1  <> object BitBtn3: TBitBtn  Left = 30  Top = 224  Width = 89  Height = 33  TabOrder = 0  Kind = bkClose end  <> object Memo1: TMemo  Left = 16  Top = 24  Width = 121  Height = 161  Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' "'  'Here a TPaintBox is ' 'used to display' 'Series Titles.'  "')  TabOrder = 1  end end End   > object OverBarForm:  TOverBarForm  Left = 187  Top = 107  Width = 598  Height = 397  Caption = 'TeeChart Overlayed  Bars Example'	T 100
end  <> object Panel1: TPanel  Left = 0  Top = 52  Width = 153  Height = 299  Align = alLeft  TabOrder = 1  <> object BitBtn3: TBitBtn  Left = 30  Top = 224  Width = 89  Height = 33  TabOrder = 0  Kind = bkClose end  <> object Memo1: TMemo  Left = 16  Top = 24  Width = 121  Height = 161  Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' "'  'Here a TPaintBox is ' 'used to display' 'Series Titles.' "')  TabOrder = 1 end end End   > object OverBarForm: TOverBarForm  Left = 187  Top = 107  Width = 598  Height = 397  Caption = 'TeeChart Overlayed Bars Example'	Top = 108
C> object Panel1: TPanel  Left = 0  Top = 52  Width = 153  Height = 299  Align = alLeft  TabOrder = 1  C> object BitBtn3: TBitBtn  Left = 30  Top = 224  Width = 89  Height = 33  TabOrder = 0  Kind = bkClose end  C> object Memo1: TMemo  Left = 16  Top = 24  Width = 121  Height = 161  Lines.Strings = (  'Chart Legends can' 'be customized using' 'any Delphi component.'  "'  'Here a TPaintBox is '  'used to display'  'Series Titles.'  "')  TabOrder = 1 end end End  End   > object OverBarForm:  TOverBarForm  Left = 187  Top = 107  Width = 598  Height = 397  Caption = 'TeeChart Overlayed Bars Example'	
Left = 0 Top = 52 Width = 153 Height = 299 Align = alLeft TabOrder = 1 <> object BitBtn3: TBitBtn Left = 30 Top = 224 Width = 89 Height = 33 TabOrder = 0 Kind = bkClose end <> object Memo1: TMemo Left = 16 Top = 24 Width = 121 Height = 161 Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' "' 'Here a TPaintBox is ' 'used to display' 'Series Titles.' "') TabOrder = 1 end end End  End   > object OverBarForm: TOverBarForm Left = 187 Top = 107 Width = 598 Height = 397 Caption = 'TeeChart Overlayed Bars Example'	7 7
Top = 52 Width = 153 Height = 299 Align = alLeft TabOrder = 1 <> object BitBtn3: TBitBtn Left = 30 Top = 224 Width = 89 Height = 33 TabOrder = 0 Kind = bkClose end <> object Memo1: TMemo Left = 16 Top = 24 Width = 121 Height = 161 Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' "' 'Here a TPaintBox is ' 'used to display' 'Series Titles.' "') TabOrder = 1 end end End  End   > object OverBarForm: TOverBarForm Left = 187 Top = 107 Width = 598 Height = 397 Caption = 'TeeChart Overlayed Bars Example'	
Width = 153  Height = 299  Align = alLeft  TabOrder = 1  <> object BitBtn3: TBitBtn  Left = 30  Top = 224  Width = 89  Height = 33  TabOrder = 0  Kind = bkClose end  <> object Memo1: TMemo  Left = 16  Top = 24  Width = 121  Height = 161  Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' "  'Here a TPaintBox is ' 'used to display' 'Series Titles.' '')  TabOrder = 1 end end End   > object OverBarForm: TOverBarForm  Left = 187  Top = 107  Width = 598  Height = 397  Caption = 'TeeChart Overlayed Bars Example'	
Height = 299 Align = alLeft TabOrder = 1 <> object BitBtn3: TBitBtn Left = 30 Top = 224 Width = 89 Height = 33 TabOrder = 0 Kind = bkClose end <> object Memo1: TMemo Left = 16 Top = 24 Width = 121 Height = 161 Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' " 'Here a TPaintBox is ' 'used to display' 'Series Titles.' '') TabOrder = 1 end end End   > object OverBarForm: TOverBarForm Left = 187 Top = 107 Width = 598 Height = 397 Caption = 'TeeChart Overlayed Bars Example'	
Align = alLeft TabOrder = 1 <> object BitBtn3: TBitBtn Left = 30 Top = 224 Width = 89 Height = 33 TabOrder = 0 Kind = bkClose end <> object Memo1: TMemo Left = 16 Top = 24 Width = 121 Height = 161 Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' "' 'Here a TPaintBox is ' 'used to display' 'Series Titles.' "') TabOrder = 1 end end End  End   > object OverBarForm: TOverBarForm Left = 187 Top = 107 Width = 598 Height = 397 Caption = 'TeeChart Overlayed Bars Example'	Width = 153
TabOrder = 1  <> object BitBtn3: TBitBtn  Left = 30  Top = 224  Width = 89  Height = 33  TabOrder = 0  Kind = bkClose end  <> object Memo1: TMemo  Left = 16  Top = 24  Width = 121  Height = 161  Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' "'  'Here a TPaintBox is ' 'used to display' 'Series Titles.' "')  TabOrder = 1 end end End  End  William	Height = 299
<pre>&lt;&gt; object BitBtn3: TBitBtn Left = 30    Top = 224    Width = 89    Height = 33    TabOrder = 0    Kind = bkClose    end    &lt;&gt; object Memo1: TMemo    Left = 16    Top = 24    Width = 121    Height = 161    Lines.Strings = (    'Chart Legends can'    'be customized using'    'any Delphi component.'    ''    'Here a TPaintBox is '    'used to display'    'Series Titles.'    '')    TabOrder = 1    end    end End  End   &gt; object OverBarForm: TOverBarForm Left = 187 Top = 107 Width = 598 Height = 397 Caption = 'TeeChart Overlayed Bars Example'</pre>	Align = alLeft
Left = 30  Top = 224  Width = 89  Height = 33  TabOrder = 0  Kind = bkClose end  <> object Memo1: TMemo  Left = 16  Top = 24  Width = 121  Height = 161  Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' ''  'Here a TPaintBox is ' 'used to display' 'Series Titles.' '')  TabOrder = 1 end end End  William And	TabOrder = 1
Left = 30  Top = 224  Width = 89  Height = 33  TabOrder = 0  Kind = bkClose end  <> object Memo1: TMemo  Left = 16  Top = 24  Width = 121  Height = 161  Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' ''  'Here a TPaintBox is ' 'used to display' 'Series Titles.' '')  TabOrder = 1 end end End  William And	<> object BitBtn3: TBitBtn
Top = 224 Width = 89 Height = 33 TabOrder = 0 Kind = bkClose end <> object Memo1: TMemo Left = 16 Top = 24 Width = 121 Height = 161 Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' '' 'Here a TPaintBox is ' 'used to display' 'Series Titles.' '') TabOrder = 1 end end End   > object OverBarForm: TOverBarForm Left = 187 Top = 107 Width = 598 Height = 397 Caption = 'TeeChart Overlayed Bars Example'	
Width = 89 Height = 33 TabOrder = 0 Kind = bkClose end <> object Memo1: TMemo Left = 16 Top = 24 Width = 121 Height = 161 Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' '' 'Here a TPaintBox is ' 'used to display' 'Series Titles.' '') TabOrder = 1 end end End William Salar Sa	
Height = 33 TabOrder = 0 Kind = bkClose end <> object Memo1: TMemo Left = 16 Top = 24 Width = 121 Height = 161 Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' "' 'Here a TPaintBox is ' 'used to display' 'Series Titles.' '') TabOrder = 1 end end End Willimminum 36 <> object OverBarForm: TOverBarForm Left = 187 Top = 107 Width = 598 Height = 397 Caption = 'TeeChart Overlayed Bars Example'	
TabOrder = 0  Kind = bkClose end  <> object Memo1: TMemo  Left = 16  Top = 24  Width = 121  Height = 161  Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' ''  'Here a TPaintBox is ' 'used to display' 'Series Titles.' '')  TabOrder = 1 end end End  End  William Safe > object OverBarForm: TOverBarForm  Left = 187  Top = 107  Width = 598  Height = 397  Caption = 'TeeChart Overlayed Bars Example'	
Kind = bkClose end  <> object Memo1: TMemo Left = 16 Top = 24 Width = 121 Height = 161 Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' '' 'Here a TPaintBox is ' 'used to display' 'Series Titles.' '') TabOrder = 1 end end End  William Safe > object OverBarForm: TOverBarForm Left = 187 Top = 107 Width = 598 Height = 397 Caption = 'TeeChart Overlayed Bars Example'	ĕ
end <> object Memo1: TMemo Left = 16 Top = 24 Width = 121 Height = 161 Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' " 'Here a TPaintBox is ' 'used to display' 'Series Titles.' ") TabOrder = 1 end end End   > object OverBarForm: TOverBarForm Left = 187 Top = 107 Width = 598 Height = 397 Caption = 'TeeChart Overlayed Bars Example'	
<> object Memo1: TMemo Left = 16 Top = 24 Width = 121 Height = 161 Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' '' 'Here a TPaintBox is ' 'used to display' 'Series Titles.' '') TabOrder = 1 end end End William San	
Left = 16 Top = 24 Width = 121 Height = 161 Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' '' 'Here a TPaintBox is ' 'used to display' 'Series Titles.' '') TabOrder = 1 end end End  End	
Top = 24 Width = 121 Height = 161 Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' '' 'Here a TPaintBox is ' 'used to display' 'Series Titles.' '') TabOrder = 1 end end End End	
Width = 121 Height = 161 Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' '' 'Here a TPaintBox is ' 'used to display' 'Series Titles.' '') TabOrder = 1 end end End	
Height = 161 Lines.Strings = ( 'Chart Legends can' 'be customized using' 'any Delphi component.' '' 'Here a TPaintBox is ' 'used to display' 'Series Titles.' '') TabOrder = 1 end end End  End	1
Lines.Strings = (     'Chart Legends can'     'be customized using'     'any Delphi component.'     ''     'Here a TPaintBox is '     'used to display'     'Series Titles.'     '')     TabOrder = 1     end     end End	
'Chart Legends can' 'be customized using' 'any Delphi component.' ''  'Here a TPaintBox is ' 'used to display' 'Series Titles.' '')  TabOrder = 1 end end End  End	
'be customized using' 'any Delphi component.'  'Here a TPaintBox is ' 'used to display' 'Series Titles.'  '')  TabOrder = 1 end end End	
'any Delphi component.'  ''  'Here a TPaintBox is '  'used to display'  'Series Titles.'  '')  TabOrder = 1  end  end  End	'Chart Legends can'
''  'Here a TPaintBox is '  'used to display'  'Series Titles.'  '')  TabOrder = 1  end  end  End	'be customized using'
'used to display' 'Series Titles.' '')  TabOrder = 1 end end End	'any Delphi component.'
'used to display' 'Series Titles.' '')  TabOrder = 1 end end End	''
'Series Titles.'  '')  TabOrder = 1  end  end  End	'Here a TPaintBox is '
'Series Titles.'  '')  TabOrder = 1  end  end  End	'used to display'
'')     TabOrder = 1     end end End	
end End  End  William Section	")
end End  End  William Section	TabOrder = 1
end  End	
End	0.1.4
36 <> object OverBarForm: TOverBarForm Left = 187 Top = 107 Width = 598 Height = 397 Caption = 'TeeChart Overlayed Bars Example'	
36 <> object OverBarForm: TOverBarForm  Left = 187  Top = 107  Width = 598  Height = 397  Caption = 'TeeChart Overlayed Bars Example'	
36 <> object OverBarForm: TOverBarForm  Left = 187  Top = 107  Width = 598  Height = 397  Caption = 'TeeChart Overlayed Bars Example'	
<> object OverBarForm: TOverBarForm  Left = 187  Top = 107  Width = 598  Height = 397  Caption = 'TeeChart Overlayed Bars Example'	
TOverBarForm  Left = 187  Top = 107  Width = 598  Height = 397  Caption = 'TeeChart Overlayed Bars Example'	
Left = 187  Top = 107  Width = 598  Height = 397  Caption = 'TeeChart Overlayed  Bars Example'	
Top = 107 Width = 598 Height = 397 Caption = 'TeeChart Overlayed Bars Example'	
Width = 598 Height = 397 Caption = 'TeeChart Overlayed Bars Example'	
Height = 397 Caption = 'TeeChart Overlayed Bars Example'	-
Caption = 'TeeChart Overlayed Bars Example'	
Bars Example'	C
•	
C I ID E	-
Color = clBtnFace	Color = clBtnFace

Title.Frame.Color = clScrollBar
Title.Text.Strings = (
'TChart Component')
Title.Visible = False
Legend.Visible = False
Align = alClient
TabOrder = 0
<> object LineSeries1:
TLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clRed
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 63
Top = 110
end
<> object LineSeries2:
TLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clGreen
AfterDrawValues =
LineSeries2AfterDrawValues
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 129

	To a City
<> object BarSeries3:	Font.Charset =
TBarSeries	DEFAULT_CHARSET
Cursor = crUpArrow	Font.Color = clBlack
Marks.ArrowLength = 20	Font.Height = -11
Marks.Visible = False	Font.Name = 'MS Sans Serif'
SeriesColor = clBlue	Font.Style = []
OnClick = BarSeries3Click	OldCreateOrder = True
BarWidthPercent = 80	Position = poScreenCenter
MultiBar = mbNone	OnCreate = FormCreate
XValues.DateTime = False	PixelsPerInch = 96
XValues.Name = 'X'	TextHeight = 13
XValues.Multiplier =	<> object Chart1: TChart
1.000000000000000000	Left = 149
XValues.Order = loAscending	Top = 0
YValues.DateTime = False	Width = 441
YValues.Name = 'Y'	Height = 363
YValues.Multiplier =	BackWall.Brush.Color = clWhite
1.000000000000000000	BackWall.Brush.Style = bsClear
YValues.Order = loNone	Foot.Font.Charset =
Left = 24	DEFAULT CHARSET
Top = 36	Foot.Font.Color = clRed
end	Foot.Font.Height = -12
<> object BarSeries2:	Foot.Font.Name = 'Arial'
TBarSeries	Foot.Font.Style = [fsItalic]
Cursor = crUpArrow	Foot.Frame.Color = clScrollBar
Marks.ArrowLength = 20	MarginTop = 2
Marks.Visible = False	Title.Brush.Color = 8421440
SeriesColor = clRed	Title.Color = 8421440
OnClick = BarSeries2Click	Title.Font.Charset =
BarWidthPercent = 60	DEFAULT CHARSET
MultiBar = mbNone	Title.Font.Color = clAqua
XValues.DateTime = False	Title.Font.Height = -19
XValues.Name = 'X'	Title.Font.Name = 'Arial'
XValues.Multiplier =	Title.Font.Style = [fsBold, fsItalic]
1.0000000000000000000000000000000000000	Title.Frame.Color = clBlue
XValues.Order = loAscending	Title.Frame.Width = 4
YValues.DateTime = False	Title.Text.Strings = (
YValues.Name = 'Y'	'Overlayed Bar Series Example')
YValues.Multiplier =	BottomAxis,Grid.Color =
1.0000000000000000000000000000000000000	clScrollBar
YValues.Order = loNone	LeftAxis.Grid.Color = clScrollBar
Left = 199	
Top = 122	Legend.Alignment = laLeft  PightAvis Crid Color =
end	RightAxis.Grid.Color = clScrollBar
<> object BarSeries1: TBarSeries	TopAxis.Grid.Color = clScrollBar
	View3D = False
Cursor = crUpArrow  Monks Arroys Longth = 20	Align = alClient
Marks.ArrowLength = 20	TabOrder = 0
Marks.Visible = False	

'OffsetPercent and'
'BarWidthPercent'
'properties of '
'TBarSeries to create'
'overlayed bars.'
"
'Bars can be displayed'
'using pattern brushes.'
")
TabOrder = 0
end
<> object BitBtn3: TBitBtn
Left = 28
Top = 316
Width = 89
Height = 33
Caption = 'Close'
TabOrder = 1
Kind = bkClose
end
<> object SpinEdit1:
TSpinEdit
Left = 75
Top = 18
Width = 50
Height = 22
Increment = 5
MaxValue = 100
MinValue = 1
TabOrder = 2
Value = 60
OnChange = SpinEdit1Change
end
<> object SpinEdit2:
TSpinEdit
Left = 75
Top = 46
Width = 50
Height = 22
Increment = 5
MaxValue = 100
MinValue = -100
TabOrder = 3
Value = 0
OnChange = SpinEdit2Change
end
<> object CBPatterns:
TCheckBox
I CHUUNDUA

SeriesColor = clLime
OnClick = BarSeries1Click
BarWidthPercent = 40
MultiBar = mbNone
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 127
Top = 46
end
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 149
Height = 363
Align = alLeft
TabOrder = 1
<> object Label1: TLabel
Left = 15
Top = 22
Width = 48
Height = 13
HelpType = htKeyword
Caption = 'Overlay %:'
end
<> object Label2: TLabel
Left = 23
Top = 46
Width = 40
Height = 13
HelpType = htKeyword
Caption = 'Offset %:'
end
<> object Memo1: TMemo
Left = 16
Top = 136
Width = 121
Height = 161
Lines.Strings = (
'This demo uses the'

### <u>AC.&P</u>

Foot.Font.Name = 'Arial'	Left = 19
Foot.Font.Style = [fsItalic]	Top = 82
Foot.Frame.Color = clScrollBar	Width = 90
Title.Frame.Color = clScrollBar	Height = 17
Title.Text.Strings = (	Caption = 'Use Patterns'
'TChart Component')	TabOrder = 4
BottomAxis.Grid.Color =	OnClick = CBPatternsClick
clScrollBar	end
Chart3DPercent = 60	end
LeftAxis.Grid.Color = clScrollBar	End
LeftAxis.Grid.Width = 0	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
RightAxis.Grid.Color =	37
clScrollBar	<> object PrintForm:
TopAxis.Grid.Color = clScrollBar	TPrintForm
TabOrder = 0	Left = 201
<> object LineSeries1:	Top = 108
TLineSeries	BorderStyle = bsDialog
Marks.ArrowLength = 8	Caption = 'TeeChart Printing
Marks.Visible = False	Demo'
SeriesColor = clRed	ClientHeight = 349
Pointer.InflateMargins = True	ClientWidth = 299
Pointer.Style = psRectangle	Color = clBtnFace
Pointer.Visible = False	ParentFont = True
XValues.DateTime = True	OldCreateOrder = True
XValues.Name = 'X'	Position = poScreenCenter
XValues.Multiplier =	Scaled = False
1.000000000000000000	OnCreate = FormCreate
XValues.Order = loAscending	PixelsPerInch = 96
YValues.DateTime = False	TextHeight = 13
YValues.Name = 'Y'	<> object Label1: TLabel
YValues.Multiplier =	Left = 9
1.000000000000000000	Top = 4
YValues.Order = loNone	Width = 72
Left = 147	Height = 13
Top = 118	HelpType = htKeyword
end	Caption = 'Enter text here:'
end This this training	end
<> object BitBtn1: TBitBtn	<> object Chart1: TChart
Left = 38	Left = 5
Top = 310	Top = 55
Width = 119	Width = 284
Height = 33	Height = 250
Caption = 'PRINT ALL !!'	BackWall.Brush.Color = clWhite
TabOrder = 1	BackWall.Brush.Style = bsClear
OnClick = BitBtn1Click	Foot.Font.Charset =
Glyph.Data = {	DEFAULT_CHARSET
	Foot.Font.Color = clRed
76010000424D76010000000000000760	Foot.Font.Height = -12
000002800000020000001000000001	

Width = 280	00
Height = 21	
TabOrder = 2	04000000000000010000120B0000120
Text = 'Hello World !'	B0000100000000000000000000000000000000
end	00
<> object BitBtn2: TBitBtn	
Left = 201	8000008000000080800080000000800
Top = 311	08000808000007F7F7F00BFBFBF00
Width = 89	0000
Height = 33	
TabOrder = 3	FF0000FF000000FFFF00FF000000F
Kind = bkClose	F00FF00FFFF0000FFFFFF0030000
end	0000000
End	
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	000337777777777777308888888888
	888807F33333333333370888888888
	88
38	88807FFFFFFFFFFFFF700000000
111111111111111111111111111111111111111	00000000777777777777777
<> object LegendXYForm:	8F8F8F
TLegendXYForm	огогог
Left = 200	8F807F333333333333F708F8F8F8F8
Top = 110	F8F9F07F333333333337370F8F8F8
Width = 569	F8F8F
Height = 370	roror
Caption = 'TeeChart Example.	8F807FFFFFFFFFFFFF700000000
<b>Customizing Legend Size and</b>	0000000077777777777773330FF
Position'	FFFFF
Color = clBtnFace	11111
ParentFont = True	03333337F3FFFF3F7F333330F0000
OldCreateOrder = True	F0F03333337F77773737F333330FFF
Position = poScreenCenter	FFFFF
OnCreate = FormCreate	
PixelsPerInch = 96	0333337F3FF3FF7F333330F00F0
TextHeight = 13	00003333337F77377773333330FFF
<> object Chart1: TChart	F0FF0
Left = 141	
Top = 0	3333337F3F37F3733333330F08F0F
Width = 420	033333337F7337F7333333330FFFF
Height = 336	0033
AllowPanning = pmNone	
AllowZoom = False	3333337FFFF773333333330000003
BackWall.Brush.Color = clWhite	3333333377777733333333}
BackWall.Brush.Style = bsClear	NumGlyphs = 2
BackWall.Pen.Visible = False	end
Foot.Frame.Color = clScrollBar	<> object Edit1: TEdit
Title.Frame.Color = clScrollBar	Left = 9
Title.Text.Strings = (	Top = 23

ID 6 1/1
'Default'
'Customized')
TabOrder = 0
OnClick = RadioGroup1Click
end
<> object BitBtn3: TBitBtn
Left = 25
Top = 272
Width = 89
Height = 33
TabOrder = 1
Kind = bkClose
end
<> object Memo1: TMemo
Left = 8
Top = 88
Width = 121
Height = 161
Lines.Strings = (
'Using the '
'OnGetLegendPos and'
'OnGetLegendRect'
'events is possible'
'to customize the Chart'
'Legend position and'
'Legend items positions.')
TabOrder = 2
end
end
End
39
<> object ScrollBarForm:
TScrollBarForm
Left = 196
Top = 108
Width = 565
Height = 331
Caption = 'TeeChart Scroll Bars
Example'
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
Scaled = False
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13

'TChart Component')
OnGetLegendPos =
Chart1GetLegendPos
OnGetLegendRect =
Chart1GetLegendRect
AxisVisible = False
ClipPoints = False
Frame.Visible = False
View3DOptions.Elevation = 315
View3DOptions.Orthogonal =
False
View3DOptions.Perspective = 0
View3DOptions.Rotation = 360
View3DWalls = False
Align = alClient
TabOrder = 0
<> object PieSeries1:
TPieSeries
Marks.ArrowLength = 8
Marks.Style = smsPercent
Marks.Visible = True
SeriesColor = clRed
OtherSlice.Text = 'Other'
PieValues.DateTime = False
PieValues.Name = 'Y'
PieValues.Multiplier =
1.00000000000000000
PieValues.Order = loNone
Left = 287
Top = 2
end
end
<> object Panel1: TPanel Left = 0
Top = 0
Width = 141
Height = 336
Align = alLeft
TabOrder = 1
<> object RadioGroup1:
TRadioGroup
Left = 14
Top = 12
Width = 103
Height = 61
Caption = 'Legend Drawing'
ItemIndex = 1
Items.Strings = (

Top = 266
Width = 523
Height = 19
PageSize = 0
TabOrder = 1
OnChange = ScrollBar1Change
end
<> object LineSeries1:
TLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clRed
Pointer.InflateMargins = True
Pointer.Style = psRectangle
Pointer.Visible = False
XValues.DateTime = True
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
Left = 199
Top = 121
end
end
End
40
<> object LogLabelsForm:
TLogLabelsForm
Left = 199
Top = 108 $Width = 631$
Height = 411
Caption = 'Logarithmic Labels
example'
Color = clBtnFace
ParentFont = True
OldCreateOrder = True
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
<> object Chart1: TChart

<> object Chart1: TChart
Left = 0
Top = 0
Width = 557
Height = 297
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
Foot.Font.Charset =
DEFAULT CHARSET
Foot.Font.Color = clRed
Foot.Font.Height = -12
Foot.Font.Name = 'Arial'
Foot.Font.Style = [fsItalic]
Foot.Frame.Color = clScrollBar
MarginBottom = 18
MarginRight = 18
Title.Frame.Color = clScrollBar
Title.Text.Strings = (
'TChart Component')
OnScroll = Chart1Scroll
OnUndoZoom =
Chart1UndoZoom
OnZoom = Chart1Zoom
BottomAxis.Grid.Color =
clScrollBar
LeftAxis.Grid.Color = clScrollBar
Legend.Visible = False
RightAxis.Grid.Color =
clScrollBar
TopAxis.Grid.Color = clScrollBar
View3D = False
Align = alClient
TabOrder = 0
OnResize = Chart1Resize
<> object ScrollBar2:
TScrollBar
Left = 517
Top = 10
•
Width = 19
Height = 255
Kind = sbVertical
PageSize = 0
TabOrder = 0
OnChange = ScrollBar2Change
end
<> object ScrollBar1:
TScrollBar
Left = 13

TabOrder = 0
Kind = bkClose
end
<> object Memo1: TMemo
Left = 12
Top = 72
Width = 121
Height = 149
Lines.Strings = (
'Logarithmic Axis'
'can draw labels in'
'decades using the'
'OnGetNextAxisLabel'
'and OnGetAxisLabel'
'events.'
**
")
TabOrder = 1
end
end
End

الملف التنفيذي لبرنامج		
(TeeChart)		
عداد		
علاء الدين اللباد		
واتف ۹۲۵۷۵۳۷۱		
برنامج يتعلق بالرسوم البيانية والإحصائية يحتوي العديد من الميزات الرائعة في برنامج دلفي ٧ و هو موجود كمثال باسم ضمن البرنامج نفسه ويمكن الاستفادة من جميع الميزات الموجودة فيه قام بتحليل وشرح البرنامج علاء الدين اللباد الملف التنفيذي للبرنامج ويحتوي على ٢٤٥ الجراء		
ProcedureS 542 وجميعها معلمة بالسهم <>		
{*************************************		

{ TeeChart Delphi Component

Left = 153
Top = 0
Width = 470
Height = 377
BackWall.Brush.Color = clWhite
BackWall.Brush.Style = bsClear
Gradient.Direction = gdLeftRight
Gradient.Visible = True
Title.Text.Strings = (
'TChart')
BottomAxis.Logarithmic = True
LeftAxis.Logarithmic = True
View3D = False
Align = alClient
TabOrder = 0
<> object Series1:
<b>TFastLineSeries</b>
ColorEachPoint = True
Marks.ArrowLength = 8
Marks.Visible = False
SeriesColor = clRed
LinePen.Color = clRed
LinePen.Width = 2
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier =
1.000000000000000000
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier =
1.000000000000000000
YValues.Order = loNone
end
end
<> object Panel1: TPanel
Left = 0
Top = 0
Width = 153
Height = 377
Align = alLeft
TabOrder = 1
<> object BitBtn2: TBitBtn
Left = 26
Top = 260
Width = 89
Height = 33
Caption = 'Close'
Сарион – Стове

Label15: TLabel;
Label16: TLabel;
Label2: TLabel;
Panel3: TPanel;
Label1: TLabel;
Label9: TLabel;
•
Label3: TLabel;
<> procedure
FormCreate(Sender: TObject);
<> procedure
Timer1Timer(Sender: TObject);
<> procedure
Button1Click(Sender: TObject);
<> procedure
Timer2Timer(Sender: TObject);
<> procedure
Button2Click(Sender: TObject);
<> procedure
Chart2MouseUp(Sender: TObject;
Button: TMouseButton;
Shift: TShiftState; X, Y:
Integer);
<> procedure
Chart2MouseDown(Sender:
TObject; Button: TMouseButton;
Shift: TShiftState; X, Y:
Integer);
<> procedure
Label10Click(Sender: TObject);
<> procedure
Button3Click(Sender: TObject);
<> procedure
Button4Click(Sender: TObject);
<> procedure
Button6Click(Sender: TObject);
<> procedure
Label3Click(Sender: TObject);
private
{ Private declarations }
DeltaZoom:Integer;
public
{ Public declarations }
<> procedure
ShowForm(FormClass:
TFormClass);
end;
Ź
var
TeeMainForm: TTeeMainForm;

<u>L</u>
Library 3.0 }
{ Demo }
{ Copyright (c) 1995-2001 by David
Berneda }
{ All rights reserved }
{***********
\ *******
unit teemain;
unit teemuni,
interface
THE TWO
uses
SysUtils, WinTypes, WinProcs,
Messages, Classes, Graphics,
Controls,
Forms, Dialogs, ExtCtrls,
TeEngine, Series, TeeProcs, Chart,
StdCtrls,
TeeFunci;
,
type
TTeeMainForm = class(TForm)
Panel1: TPanel;
Button1: TButton;
Image1: TImage;
Timer1: TTimer;
Button2: TButton;
Button3: TButton;
Button4: TButton;
Button6: TButton;
Chart2: TChart;
FastLineSeries1: TFastLineSeries;
FastLineSeries2: TFastLineSeries;
Timer2: TTimer;
Label11: TLabel;
Label12: TLabel;
Panel2: TPanel;
Chart1: TChart;
Series1: TPieSeries;
Chart7: TChart;
PieSeries6: TPieSeries;
Chart8: TChart;
PieSeries7: TPieSeries;
Chart9: TChart;
PieSeries8: TPieSeries;
Chart10: TChart;
PieSeries9: TPieSeries;
Label13: TLabel;

PieSeries7.Rotate(5);	
PieSeries9.Rotate(5);	implementation
Series1.Rotate(355);	Implementation
PieSeries8.Rotate(355);	(\$D * dfm)
With Chart9.View3DOptions do	{\$R *.dfm}
	Hass Too About Too Dasia Footuuss
begin	Uses TeeAbout, TeeBasic, Features,
Zoom:=Zoom-DeltaZoom;	UDemoCha, Specs, ShellAPI;
if (Zoom<60) or (Zoom>110) then	
DeltaZoom:=-DeltaZoom;	<> procedure
end;	TTeeMainForm.ShowForm(FormCl
With PieSeries8 do	ass: TFormClass);
begin	begin
tmp:=Random(Count);	Timer1.Enabled:=False;
if ExplodedSlice[tmp]>20 then	Timer2.Enabled:=False;
	With FormClass.Create(Self) do
ExplodedSlice[tmp]:=ExplodedSlice[	try
tmp]-1	ShowModal;
else	finally
	Free;
ExplodedSlice[tmp]:=ExplodedSlice[	end;
tmp]+1;	Timer2.Enabled:=True;
end;	Timer1.Enabled:=True;
end;	end;
,	
<> procedure	<> procedure
TTeeMainForm.Button1Click(Sende	TTeeMainForm.FormCreate(Sender
r: TObject);	: TObject);
begin	begin
Close;	TeeEraseBack:=False;
end;	PieSeries6.FillSampleValues(5);
cnu,	<u> </u>
<> procedure	PieSeries7 FillSampleValues(5);
TTeeMainForm.Timer2Timer(Send	PieSeries7.FillSampleValues(5);
er: TObject);	Series1.FillSampleValues(5);
	PieSeries8.CheckDataSource;
var tmpX:Double;	
begin	FastLineSeries1.FillSampleValues(2
With FastLineSeries1 do	0);
begin	
tmpX:=XValues[1]-XValues[0];	FastLineSeries2.FillSampleValues(2
Delete(0);	0);
	DeltaZoom:=5;
AddXY(XValues.Last+tmpX,Rando	end;
m(100),",clTeeColor);	
end;	<> procedure
With FastLineSeries2 do	TTeeMainForm.Timer1Timer(Send
begin	er: TObject);
tmpX:=XValues[1]-XValues[0];	var tmp:Integer;
Delete(0);	begin
(-);	PieSeries6.Rotate(5);
	11000110001100000(0))

ShowForm(TChartSpecs);	AddXY(XValues.Last+tmpX,Rando
	m(100),",clTeeColor);
end;	
c Summer dans	end; end;
<pre>&gt; procedure TTooMainForm Putton(Click(Sanda))</pre>	enu;
TTeeMainForm.Button6Click(Sende	
r: TObject);	<pre>&lt;&gt; procedure TT M -in Ferman Posts 20th (See Asset)</pre>
begin CTC : F	TTeeMainForm.Button2Click(Sende
ShowForm(TSeriesForm);	r: TObject);
end;	begin ShowForm(TDemoForm);
Z. Nama a Jama	
<pre>&lt;&gt; procedure TTooMoinForm Lobel2Click(Sondon)</pre>	end;
TTeeMainForm.Label3Click(Sender	
: TObject);	<pre>&lt;&gt; procedure TTooMainForm Chart2MangeUn(See</pre>
Var St:Array[0255] of char;	TTeeMainForm.Chart2MouseUp(Se
begin Ti 1 F 1 1 1 F 1	nder: TObject; Button:
Timer1.Enabled:=False;	TMouseButton; Shift: TShiftState; X, Y: Integer);
Timer2.Enabled:=False;	, , ,
ShallEvacuto(0.1anan! Str.DCana(St.)	begin Label10Click(Solf)
ShellExecute(0,'open',StrPCopy(St,'	Label10Click(Self);
http://'+Label3.Caption+	end;
!/nmoduats/taashant/dalnhi6!) nil nil S	< > nuacaduna
'/products/teechart/delphi6'),nil,nil,S W SHOW);	<> procedure TTeeMainForm.Chart2MouseDown
<del>-</del> ···	
end;	(Sender: TObject; Button: TMouseButton; Shift:
end.	TShiftState; X, Y: Integer);
ciiu.	begin
	Chart2.BevelOuter:=bvLowered;
	end;
program Teedemo;	cnu,
program recuemo,	<> procedure
uses	TTeeMainForm.Label10Click(Sende
Forms,	r: TObject);
teemain in 'teemain.PAS'	begin
{TeeMainForm},	ShowForm(TTeeAboutForm);
Teebasic in 'teebasic.PAS'	Chart2.BevelOuter:=bvRaised;
{DemoForm},	end;
Features in 'features.pas'	
{FeaturesForm},	<> procedure
Udemocha in 'UDemoCha.pas'	TTeeMainForm.Button3Click(Sende
{SeriesForm},	r: TObject);
Basic in 'basic.pas' {BasicForm},	begin
Bubble in 'bubble.pas'	ShowForm(TFeaturesForm);
{BubbleForm},	end;
Gantt in 'Gantt.pas' {GanttForm},	
Lastvalu in 'Lastvalu.pas'	<> procedure
{DigitalForm},	TTeeMainForm.Button4Click(Sende
Linked in 'Linked.pas'	r: TObject);
{LinkedTablesForm},	begin
(),	~~ <del>5</del> ~~~

Ushapes in 'Ushapes.pas'
{ShapesForm},
Ustack in 'Ustack.pas'
{StackedForm},
Uylegend in 'Uylegend.pas'
{LegendXYForm};
{\$R *.RES}
begin
Application.Initialize;
Application.CreateForm(TTeeMain Form, TeeMainForm);
Application.Run;
end.
{*************************************
******
{ TeeChart Delphi Component
Library }
{Areas Demo }
{ Copyright (c) 1995-2001 by David
Berneda }
{ All rights reserved } {***********************************
<b>******</b>
unit stackare;
unit stackare,
interface
interrace
{ This form shows 3 TAreaSeries
components on same Chart.
Areas can be Stacked or Stacked
100%
}
uses
SysUtils, WinTypes, WinProcs,
Messages, Classes, Graphics,
Controls,
Forms, Dialogs, StdCtrls, Teengine,
Series, Buttons, Chart, ExtCtrls,
TeeProcs
•
7
type
TAreasForm = class(TForm)
` ,
Panel1: TPanel;

LogLab in 'loglab.pas'
{LogLabelsForm},
Mulaxis in 'mulaxis.pas'
{CustomAxisForm},
Pie in 'Pie.pas' {PieForm},
Specs in 'Specs.pas' {ChartSpecs},
Sqlbars in 'Sqlbars.pas'
{SQLBarsForm},
Stackare in 'stackare.pas'
{AreasForm},
Tablepie in 'Tablepie.pas'
{TablePieForm},
Uanizoom in 'Uanizoom.pas'
{FormAnimatedZoom},
Uarrows in 'uarrows.pas'
{ArrowsForm},
Uaxislab in 'Uaxislab.pas'
{AxisLabelsForm},
UBitmap in 'Ubitmap.pas'
{BitmapForm},
ucolor in 'ucolor.pas'
{ColoredForm},
Ucrossh in 'ucrossh.pas'
{CrossHairForm},
Udbhoriz in 'Udbhoriz.pas'
{DBHorizBarForm},
udemutil in 'Udemutil.pas',
Udraw in 'Udraw.pas'
{DrawForm},
UFast in 'Ufast.pas'
{FastLineForm},
UHighLo in 'uhighlo.pas'
{HighLowForm},
Ukeyboa in 'ukeyboa.pas'
{KeyboardForm},
Ulegend in 'Ulegend.pas'
{LegendForm},
Umain in 'Umain.pas' {MiniForm},
UMetafil in 'umetafil.pas'
{MetafileForm},
Uoverbar in 'Uoverbar.pas'
{OverBarForm},
Upages in 'Upages.pas'
{PagesForm},
Uprint in 'Uprint.pas' {PrintForm},
Uscroll in 'Uscroll.pas'
{ScrollForm},
Uscrollb in 'Uscrollb.pas'
{ScrollBarForm},

XX7° (X A X	D. I. C 1 ED. I. C.
With Area do	RadioGroup1: TRadioGroup;
begin	Chart1: TChart;
XValues.DateTime:=False;	BitBtn1: TBitBtn;
Clear;	AreaSeries1: TAreaSeries;
Old:=500+Random(1000);	AreaSeries2: TAreaSeries;
for t:=1 to 30 do	AreaSeries3: TAreaSeries;
begin	CheckBox1: TCheckBox;
Old:=Old+Random(50)-25;	Label1: TLabel;
Add( Old,",clTeeColor);	CheckBox2: TCheckBox;
end;	Timer1: TTimer;
Cursor:=crTeeHand;	<> procedure
end;	FormCreate(Sender: TObject);
end;	<> procedure
	RadioGroup1Click(Sender:
begin	TObject);
tmpCounter:=-1;	<> procedure
tmpCounter:1; tmpSeries:=nil;	CheckBox1Click(Sender: TObject);
•	<> procedure
CreateRandom(AreaSeries1);	Chart1MouseMove(Sender:
CreateRandom(AreaSeries2);	`
CreateRandom(AreaSeries3);	TObject; Shift: TShiftState; X,
end;	Y: Integer);
	<> procedure
<> procedure	CheckBox2Click(Sender: TObject);
TAreasForm.RadioGroup1Click(Se	<> procedure
nder: TObject);	Timer1Timer(Sender: TObject);
begin	private
{ Change how areas are displayed.	{ Private declarations }
(Stacked, Stacked 100%) }	public
	{ Public declarations }
AreaSeries1.MultiArea:=TMultiAre	
a(RadioGroup1.ItemIndex);	tmpDelta,tmpCounter,tmpIndex:Lo
end;	ngint;
	tmpSeries:TAreaSeries;
<> procedure	end;
TAreasForm.CheckBox1Click(Send	
er: TObject);	implementation
begin	•
	{\$R *.dfm}
Chart1.View3D:=CheckBox1.Check	
ed; { < turn on/off 3d }	{ Some random values }
end;	<> procedure
ciru,	TAreasForm.FormCreate(Sender:
<> procedure	TObject);
TAreasForm.Chart1MouseMove(Se	i object),
`	<> procedure
nder: TObject; Shift: TShiftState;	CreateRandom(Area:TAreaSeries);
X, Y: Integer);	,
- Samuel 1	var t:Longint;
<> procedure	Old:Double;
HitSeries(ASeries:TChartSeries);	begin

With tmpSeries.GetVertAxis do	Var tmp:Longint;
tmpDelta:=Round(Abs(Maximum-	begin
Minimum)/50.0);	tmp:=ASeries.Clicked(x,y);
if Random(2)=1 then tmpDelta:=-	if tmp<>-1 then
tmpDelta;	begin
end;	Label1.Caption:=ASeries.Name;
inc(tmpCounter);	•
	Label1.Font.Color:=ASeries.SeriesC
tmp:=tmpSeries.YValue[tmpIndex];	olor;
if (tmp+tmpDelta)>0 then	end;
tmpSeries.YValue[tmpIndex]:=tmp+	end;
tmpDelta;	
end;	var t:Longint;
	begin
end.	Label1.Caption:=";
111111111111111111111111111111111111111	for t:=0 to Chart1.SeriesCount-1 do
	HitSeries(Chart1.Series[t]);
3	
{**********	Label1.Visible:=Label1.Caption<>'';
*******	end;
{ TeeChart Delphi Component	onu,
Library }	<> procedure
{ TArrowSeries Example	TAreasForm.CheckBox2Click(Send
}	er: TObject);
{ Copyright (c) 1995-2001 by David	begin
Berneda }	begin
{ All rights reserved }	Timer1.Enabled:=CheckBox2.Check
{*************************************	ed;
*******	if Timer1.Enabled then
unit uarrows;	RadioGroup1.ItemIndex:=1;
	end;
interface	onu,
	<> procedure
uses	TAreasForm.Timer1Timer(Sender:
SysUtils, WinTypes, WinProcs,	TObject);
Messages, Classes, Graphics,	var tmp:Double;
Controls,	begin
Forms, Dialogs, Teengine, Series,	if (tmpCounter=-1) or
ArrowCha, Chart, StdCtrls,	(tmpCounter=5) then
ExtCtrls,	begin
Buttons, TeeProcs;	Case Random(3) of
Dations, 1001100s,	
tyne	0: tmpSeries:=AreaSeries1;
type TArrowsForm = class(TForm)	1: tmpSeries:=AreaSeries2;
	2: tmpSeries:=AreaSeries3;
Panel1: TPanel;	end;
CheckBox1: TCheckBox;	
Chart1: TChart;	tmpIndex:=Random(tmpSeries.Cou
ArrowSeries1: TArrowSeries;	nt);
Timer1: TTimer;	tmpCounter:=0;

<del>-</del>	10.001
x1:=Random( 300 ) - 150.0;	] [
if x1<50 then x1:=50;	
x1:=x1+x0;	
,	Fo
y1:=Random( 300 ) - 150.0;	1
if y1<50 then y1:=50;	C
y1:=y1+y0;	1
J =	Ti
AddArrow( x0,y0,x1,y1, ",	
clTeeColor);	
end;	
end;	1 -
end;	1
	$\mathbf{A}$
<> procedure	6
TArrowsForm.CheckBox1Click(Sen	
der: TObject);	in
begin	1
	{\$
Timer1.Enabled:=CheckBox1.Check	
ed;	<-
end;	$\mathbf{T}$
	T
<> procedure	be
TArrowsForm.Timer1Timer(Sender	
: TObject);	I
var t:Longint;	1
begin	1
Timer1.Enabled:=False;	1
With ArrowSeries1 do	
Begin	
for t:=0 to Count-1 do	
Begin	1
- 8	
StartXValues[t]:=StartXValues[t]+R	er
andom(100)-50.0;	
	<-
StartYValues[t]:=StartYValues[t]+R	$  T_{\perp}$
andom(100)-50.0;	Va
EndXValues[t]	1
:=EndXValues[t]+Random(100)-	be
50.0;	
EndYValues[t]	
:=EndYValues[t]+Random(100)-	
50.0;	<u> </u>
End;	]  -
Repaint;	] <del> </del>
End;	]
Timer1.Enabled:=True;	
·	-

BitBtn3: TBitBtn;
Memo1: TMemo;
<> procedure
FormCreate(Sender: TObject);
<> procedure
CheckBox1Click(Sender: TObject);
<> procedure
Timer1Timer(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
<> procedure
AddRandomArrows;
end;
implementation
{\$R *.dfm}
<> procedure
TArrowsForm.FormCreate(Sender:
TObject);
begin
With ArrowSeries1 do
Begin
ArrowWidth:=32;
ArrowHeight:=24;
ColorEachPoint:=True;
XValues.DateTime:=False;
YValues.DateTime:=False;
4 1170 1 4
AddRandomArrows;
end;
end;
<> procedure
TArrowsForm.AddRandomArrows;
var x0,y0,x1,y1:Double;
t:Longint;
begin
With ArrowSeries1 do
Begin
Clear;
for t:=1 to 40 do
begin (1999)
x0:=Random( 1000 );
y0:=Random( 1000 );

	1
any Font size adjustment.	end;
TeeChart Axis will draw all labels	
you specify.	end.
}	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
uses	4
SysUtils, WinTypes, WinProcs,	{***********************
Messages, Classes, Graphics,	******
Controls,	{ TeeChart. TChart Component
Forms, Dialogs, Chart, Series,	}
ExtCtrls, StdCtrls, Teengine,	{ Copyright (c) 1995-2001 by David
Buttons,	Berneda }
TeeProcs;	{ All Rights Reserved }
	{*************************************
type	*******
TAxisLabelsForm = class(TForm)	unit uaxislab;
Chart1: TChart;	,
LineSeries1: TLineSeries;	interface
Panel1: TPanel;	A110111100
RadioGroup1: TRadioGroup;	{ This Sample Project shows how to
PointSeries1: TPointSeries;	set the Axis Labels at specific Axis
BitBtn3: TBitBtn;	positions.
, ,	1
Memo1: TMemo;	The key is the
<> procedure	Chart.OnGetNextAxisLabel event.
FormCreate(Sender: TObject);	This event is called continuouly for
<> procedure	each Axis Label until user decides
RadioGroup1Click(Sender:	to stop.
TObject);	At each call, you can specify the
<> procedure	exact Axis value where a Label must
Chart1GetNextAxisLabel(Sender:	be
TChartAxis;	drawn.
LabelIndex: Longint; var	
LabelValue: Double; var Stop:	In this example, this event is used
Boolean);	to set the BottomAxis labels to the
private	first (1) day in month: 1/1/96,
{ Private declarations }	2/1/96, 3/1/96 12/1/96
public	
{ Public declarations }	This don't needs necessarily to be
DefaultLabels:Boolean;	datetime values.
end;	You can also set the Axis Labels in
,	non-datetime axis.
implementation	
F	WARNING:
{\$R *.dfm}	Remember to set the Stop boolean
(*** '*****)	variable to TRUE when no more
<> procedure	labels are
<u>-</u>	needed.
TAxisLabelsForm.FormCreate(Send	Remember also that using this
er: TObject);	event will NOT calculate Label used
var t:Longint;	
begin	space or

Chart1.BottomAxis.Increment	DefaultLabels:=False; { <
	boolean variable to show or not the
:= DateTimeStep[ dtOneMonth ];	
1	demo }
and	LingCopies 1 Cleans
	LineSeries1.Clear;
	PointSeries1.Clear;
Chart1.BottomAxis.ExactDateTime	for t:=1 to 100 do
:= True ;	Begin
	LineSeries1.AddXY( Date+t,
Eliminates the need for the	200+Random(700),",clTeeColor); {
following code.	< some random points }
}	PointSeries1.AddXY( Date+t,
\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	200+Random(700),",clTeeColor);
***************************************	end;
	end;
{ LabelValue has the "candidate"	
value where the Axis label will be	<> procedure
painted. }	TAxisLabelsForm.RadioGroup1Clic
	k(Sender: TObject);
DecodeDate(LabelValue, year, month,	begin
day);	{ Choose between default and
{ we force that value to be the	custom labeling. }
first day in month }	
Day:=1;	DefaultLabels:=RadioGroup1.ItemI
Month:=Month+1;	ndex=0;
if Month>12 then	Chart1.Repaint; { < repaint chart
Begin	to see changes }
Month:=1;	end;
Year:=Year+1;	
end;	<> procedure
{ Then we set the preferred	TAxisLabelsForm.Chart1GetNextA
Label value }	xisLabel(Sender: TChartAxis;
	LabelIndex: Longint; var
LabelValue:=EncodeDate(year,mont	LabelValue: Double; var Stop:
h,day);	Boolean);
end	var year,month,day:Word;
else	begin
if Sender=Chart1.LeftAxis then	if not DefaultLabels then
Begin	Begin
{ In this example, we want the	if Sender=Chart1.BottomAxis
Vertical Left Axis to show	then
labels only for positive values,	Begin
starting at zero and	{
with 250 label increment.	**********
}	{ WARNING:
if LabelValue>=250 then	
LabelValue:=LabelValue+250	Setting this axis increment:
else LabelValue:=250;	
220 220 1 1 1 1 2 2 3 9	<u> </u>

_
Writing: TPointSeries;
Reading: TPointSeries;
BitBtn1: TBitBtn;
RadioGroup1: TRadioGroup;
BitBtn3: TBitBtn;
CheckBox2: TCheckBox;
SaveDialog1: TSaveDialog;
BitBtn2: TBitBtn;
Winter: TLineSeries;
Summer: TLineSeries;
<> procedure
FormCreate(Sender: TObject);
<> procedure
CheckBox1Click(Sender: TObject);
<> procedure
Timer1Timer(Sender: TObject);
<> procedure
FormResize(Sender: TObject);
<> procedure
FormShow(Sender: TObject);
<> procedure
BitBtn1Click(Sender: TObject);
<> procedure
CheckBox2Click(Sender: TObject);
<> procedure
BitBtn2Click(Sender: TObject);
private (Sprivate desloyations)
{ Private declarations }
public
{ Public declarations }
end;
implementation
{\$R *.dfm}
Uses Printers;
,
<> procedure
TBasicForm.FormCreate(Sender:
TObject);
var t:Longint;
begin
G
Summer.FillSampleValues(20); { <-
- Some random points }
Winter.FillSampleValues(20); { <
Some random points }
{ Will need a Bar Series with

End;	
{ we want more labels !! }	
Stop:=False;	
end;	
end;	
end.	
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	
5	
{************	
\ \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	
{ TeeChart Delphi Component	
Library }	
{ Basic Series Types Demo	
{ Dasic Series Types Delliu	
{ Copyright (c) 1995-2001 by David	
{ Copyright (c) 1995-2001 by David   Berneda   }	
,	
{ All rights reserved }	
\{\`\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	
<u> </u>	
unit basic;	
•	
interface	
uses	
SysUtils, WinTypes, WinProcs,	
Messages, Classes, Graphics,	
Controls,	
Forms, Dialogs, Chart, Series,	
ExtCtrls, StdCtrls, Bubble,	
BubbleCh, Teengine, Buttons,	
teeprocs;	
type	
TBasicForm = class(TForm)	
Chart1: TChart;	
Panel1: TPanel;	
CheckBox1: TCheckBox;	
Timer1: TTimer;	
BarSeries1: TBarSeries;	
South: TAreaSeries;	
Speaking: TPointSeries;	
Panel2: TPanel;	
Chart2: TChart;	
1 (114114, 1(11411)	
,	
Label1: TLabel;	
Label1: TLabel; Chart3: TChart;	
Label1: TLabel;	

",clTeeColor);	special random values,
end;	so we dont call the standard
end;	FillSampleValues method }
enu,	With BarSeries1 do
Von ton LabaliStrings	for t:=1 to 12 do
Var tmpLabel:String;	
tmpColor:TColor;	Add(Random(1000),ShortMonthNa
begin	mes[t],GetDefaultColor(t));
Animate(Summer);	South EillSompleValues(20): ( <
Animate(Winter);	South.FillSampleValues(20); { <
{ BarSeries1 has special treatment	Some random points }
to animate }	North.FillSampleValues(20); { <
With BarSeries1 do	Some random points }
Begin	G II FING I VI (A0) (
tmpLabel:=XLabel[0];	Speaking.FillSampleValues(20); {
tmpColor:=ValueColor[0];	< Some random points }
Delete(0);	Reading.FillSampleValues(20); { <-
	- Some random points }
AddXY(XValues.Last+1,Random(10	Writing.FillSampleValues(20); { <
00),tmpLabel,tmpColor);	Some random points }
{ Change Bar Style randomly }	{ Force to resize the four charts }
if Random(10)<1 then	FormResize(Self);
	end;
BarStyle:=TBarStyle(Random(1+Or	
d(High(TBarStyle))));	<> procedure
end;	TBasicForm.CheckBox1Click(Sende
Animate(South);	r: TObject);
Animate(North);	begin
Animate(Speaking);	
Animate(Reading);	Timer1.Enabled:=CheckBox1.Check
Animate(Writing);	ed;
	end;
{ Change Pointer Style randomly	
}	<> procedure
if Random(10)<1 then	TBasicForm.Timer1Timer(Sender:
With Speaking do	TObject);
With Speaking uv	* //
Pointer.Style:=TSeriesPointerStyle(	<> procedure
Random(1+Ord(High(TSeriesPointe	Animate(Series:TChartSeries);
rStyle))));	Begin
	With Series do
end;	Begin
( Dealign the four shorts )	Delete(0); { < remove the first
{ Realign the four charts }	point }
<pre>&lt;&gt; procedure TPasia Farry Farry Pasia (Sandary)</pre>	{ Add a new random point }
TBasicForm.FormResize(Sender:	AddXY( XValues.Last+1,
TObject);	AuuA1(Avaiucs.Last 1,
Var w,h:Longint;	VValues I ast+(Dandam(ChautSamn
begin	YValues.Last+(Random(ChartSamp
{ Top and bottom Panel positioning	lesMax)-(ChartSamplesMax/2)),

{ < Force Horizontal paper }	}
try	w:=ClientWidth-4-Panel1.Width;
Printer.BeginDoc; { < start	Panel2.Width:=w-4;
printer job }	Panel1.Height:=ClientHeight-
try	Panel2.Height-4;
Printer.Title:='TeeChart	h:=Panel1.Height;
Printing Demo';	{ Top Left Chart }
Case RadioGroup1.ItemIndex of	Chart1.SetBounds(Panel1.Width,Pa
0: Begin { screen proportional }	nel2.Height,w div 2,h div 2);
Chart1.PrintResolution:= 0;	{ Bottom Left Chart }
Chart2.PrintResolution:= 0;	
Chart3.PrintResolution:= 0;	Chart2.SetBounds(Panel1.Width,Pa
Chart4.PrintResolution:= 0;	nel2.Height+Chart1.Height,w div 2,h
End;	div 2);
1: Begin { thin lines and small	{ Top Right Chart }
fonts }	Chaut? CatDourds (Donald Wildel C
Chart1.PrintResolution:= -	Chart3.SetBounds(Panel1.Width+C hart1.Width+1,Chart1.Top,w div 2,h
100;	div 2);
Chart2.PrintResolution:= -	{ Bottom Right Chart }
Chart3.PrintResolution:= -	{ Bottom Right Chart }
100;	Chart4.SetBounds(Chart3.Left,Char
Chart4.PrintResolution:= -	t2.Top,w div 2,h div 2);
100;	end;
End;	
end;	<> procedure
	TBasicForm.FormShow(Sender:
{ Print the four charts, each one	TObject);
at a different paper position }	begin
, , , , , , , , , , , , , , , , , , , ,	FormResize(Self); { < to align
{ CALCULATE HORIZONTAL	charts }
MARGIN }	end;
tmpW:=Printer.PageWidth;	
	<> procedure
tmpWMargin:=Round(5.0*tmpW/10	TBasicForm.BitBtn1Click(Sender:
0.0); { < 5% margins }	TObject);
tmpW:=tmpW-2*tmpWMargin;	Var
{ < left and right margins }	tmpH,tmpW,tmpWMargin,tmpHM
tmpW:=tmpW div 2; { half	argin:Longint; { margins }
height for left and right charts }	OldOwiewtek TD-i( O i / / )
	OldOrientation:TPrinterOrientation
{ CALCULATE VERTICAL	hogin
MARGIN }	begin Screen.Cursor := crHourGlass;
tmpH:=Printer.PageHeight;	Screen.Cursor :- criticurglass;
Annual De March	OldOrientation:=Printer.Orientatio
tmpHMargin:=Round(5.0*tmpH/10	n; { < save paper orientation }
0.0); { < 5% margins } tmpH:=tmpH-2*tmpHMargin; {	Printer.Orientation:=poLandscape;
	I I IIIICI .OI ICHIAUUHDULAHUSCADE,

wastowa awasaw)	c hottom and ton manning)
restore cursor }	< bottom and top margins }
end;	tmpH:=tmpH div 2; { half height
end;	for top and bottom charts }
<> procedure	{ left / top chart }
TBasicForm.CheckBox2Click(Sende	Chart1.PrintPartial( Rect(
r: TObject);	tmpWMargin,tmpHMargin,
begin	
	tmpWMargin+tmpW,tmpHMargin+
Chart1.View3D:=CheckBox2.Check	tmpH ) );
ed;	
	{ right / top chart }
Chart2.View3D:=CheckBox2.Check	Chart3.PrintPartial( Rect(
ed;	tmpWMargin+tmpW,tmpHMargin,
Chart3.View3D:=CheckBox2.Check	tmpWMargin+2*tmpW,tmpHMargi
ed;	n+tmpH));
	1 ///
Chart4.View3D:=CheckBox2.Check	{ left / bottom chart }
ed;	Chart2.PrintPartial( Rect(
end;	tmpWMargin,tmpHMargin+tmpH,
chu,	tinp wiviai gin,tinpiiiviai gin tinpii,
<> procedure	tmpWMargin+tmpW,tmpHMargin+
TBasicForm.BitBtn2Click(Sender:	2*tmpH));
TObject);	2 tmp11 ) ),
var tmpH,tmpW:Longint;	(vight / hottom short)
begin	{ right / bottom chart }
	Chart4.PrintPartial( Rect(
{ This code creates and stores a new BITMAP file	tmpWMargin+tmpW,tmpHMargin+
	tmpH,
which contains the FOUR charts.	4
Asks previously the user the BMP	tmpWMargin+2*tmpW,tmpHMargi
filename.	n+2*tmpH));
}	Did E ID (4 111
with SaveDialog1 do	Printer.EndDoc; { < end job
begin	and print!!}
if Execute then	except
With TBitmap.Create do	on Exception do { just in case an
try	error happens }
{ calculate bitmap size (2x2) }	Begin
tmpW:=Chart1.Width;	Printer.Abort;
tmpH:=Chart1.Height;	Printer.EndDoc;
Width := 2*tmpW;	Raise;
Height:= 2*tmpH;	end;
{ draw chart 1 }	end;
Chart1.BufferedDisplay:=False;	finally
Chart1.Draw(Canvas,Rect(0,0,tmp	Printer.Orientation:=OldOrientatio
W,tmpH));	n; { < restore paper orientation }
Chart1.BufferedDisplay:=True;	Screen.Cursor:=crDefault; { <

{ Bitmap images (TBitmap	
component) can be painted in	{ draw chart 2 }
several styles:	Chart2.BufferedDisplay:=False;
Several Segress	Chartz.buffercubisplay. Taise,
pbmStretch : Bitmap is stretched to	Chart2.Draw(Canvas,Rect(0,tmpH+
fit Chart Panel Rectangle	1,tmpW,2*tmpH));
pbmTile: Bitmap is repeteadly	Chart2.BufferedDisplay:=True;
painted without stretching.	Chartz.Bullercubisplay. 11 uc,
pbmCenter: Bitmap is painted	{ draw chart 3 }
centered without stretching.	Chart3.BufferedDisplay:=False;
centered without streteming.	Charts. Burier cubispiay.—Paise,
You use the PanelBitmapMode	Chart3.Draw(Canvas,Rect(tmpW+1,
property:	0,2*tmpW,tmpH));
property	Chart3.BufferedDisplay:=True;
Chart1.PanelBitmapMode :=	Charto. Duffer cubispiay. 11 uc,
pbmTile;	{ draw chart 4 }
}	Chart4.BufferedDisplay:=False;
,	Chart-Builer cubisplay. Taise,
uses	Chart4.Draw(Canvas,Rect(tmpW+1,
SysUtils, WinTypes, WinProcs,	tmpH+1,2*tmpW,2*tmpH));
Messages, Classes, Graphics,	Chart4.BufferedDisplay:=True;
Controls,	SaveToFile(FileName);
Forms, Dialogs, ExtCtrls, Chart,	finally
Series, StdCtrls, Teengine, Buttons,	Free;
TeeProcs;	end;
,	
	end;
type	
type TBitmapForm = class(TForm)	end; end;
type TBitmapForm = class(TForm) Chart1: TChart;	end;
type TBitmapForm = class(TForm) Chart1: TChart; OpenDialog1: TOpenDialog;	end; end; end.
type TBitmapForm = class(TForm) Chart1: TChart; OpenDialog1: TOpenDialog; Panel1: TPanel;	end; end; end.
type TBitmapForm = class(TForm) Chart1: TChart; OpenDialog1: TOpenDialog; Panel1: TPanel; RadioGroup1: TRadioGroup;	end; end; end.
type TBitmapForm = class(TForm) Chart1: TChart; OpenDialog1: TOpenDialog; Panel1: TPanel; RadioGroup1: TRadioGroup; BitBtn1: TBitBtn;	end; end; end.
type TBitmapForm = class(TForm) Chart1: TChart; OpenDialog1: TOpenDialog; Panel1: TPanel; RadioGroup1: TRadioGroup; BitBtn1: TBitBtn; BitBtn3: TBitBtn;	end; end.  ()))))))))))))))))))))))))))))))))))
type TBitmapForm = class(TForm) Chart1: TChart; OpenDialog1: TOpenDialog; Panel1: TPanel; RadioGroup1: TRadioGroup; BitBtn1: TBitBtn; BitBtn3: TBitBtn; CheckBox1: TCheckBox;	end; end; end.
type TBitmapForm = class(TForm) Chart1: TChart; OpenDialog1: TOpenDialog; Panel1: TPanel; RadioGroup1: TRadioGroup; BitBtn1: TBitBtn; BitBtn3: TBitBtn; CheckBox1: TCheckBox; Series1: TBarSeries;	end; end.
type TBitmapForm = class(TForm) Chart1: TChart; OpenDialog1: TOpenDialog; Panel1: TPanel; RadioGroup1: TRadioGroup; BitBtn1: TBitBtn; BitBtn3: TBitBtn; CheckBox1: TCheckBox; Series1: TBarSeries; Button1: TButton;	end; end.
type  TBitmapForm = class(TForm)  Chart1: TChart;  OpenDialog1: TOpenDialog;  Panel1: TPanel;  RadioGroup1: TRadioGroup;  BitBtn1: TBitBtn;  BitBtn3: TBitBtn;  CheckBox1: TCheckBox;  Series1: TBarSeries;  Button1: TButton;  Memo1: TMemo;	end; end.   \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
type  TBitmapForm = class(TForm)  Chart1: TChart;  OpenDialog1: TOpenDialog;  Panel1: TPanel;  RadioGroup1: TRadioGroup;  BitBtn1: TBitBtn;  BitBtn3: TBitBtn;  CheckBox1: TCheckBox;  Series1: TBarSeries;  Button1: TButton;  Memo1: TMemo;  <> procedure	end; end.
type  TBitmapForm = class(TForm)  Chart1: TChart;  OpenDialog1: TOpenDialog;  Panel1: TPanel;  RadioGroup1: TRadioGroup;  BitBtn1: TBitBtn;  BitBtn3: TBitBtn;  CheckBox1: TCheckBox;  Series1: TBarSeries;  Button1: TButton;  Memo1: TMemo;  <> procedure  FormCreate(Sender: TObject);	end; end.  end.  ()()()()()()()()()()()()()()()()()()(
type  TBitmapForm = class(TForm)  Chart1: TChart;  OpenDialog1: TOpenDialog;  Panel1: TPanel;  RadioGroup1: TRadioGroup;  BitBtn1: TBitBtn;  BitBtn3: TBitBtn;  CheckBox1: TCheckBox;  Series1: TBarSeries;  Button1: TButton;  Memo1: TMemo;  <> procedure  FormCreate(Sender: TObject);  <> procedure	end; end.  end.  ()()()()()()()()()()()()()()()()()()(
type  TBitmapForm = class(TForm)  Chart1: TChart;  OpenDialog1: TOpenDialog;  Panel1: TPanel;  RadioGroup1: TRadioGroup;  BitBtn1: TBitBtn;  BitBtn3: TBitBtn;  CheckBox1: TCheckBox;  Series1: TBarSeries;  Button1: TButton;  Memo1: TMemo;  <> procedure  FormCreate(Sender: TObject);  <> procedure  RadioGroup1Click(Sender:	end; end.
type  TBitmapForm = class(TForm)  Chart1: TChart;  OpenDialog1: TOpenDialog;  Panel1: TPanel;  RadioGroup1: TRadioGroup;  BitBtn1: TBitBtn;  BitBtn3: TBitBtn;  CheckBox1: TCheckBox;  Series1: TBarSeries;  Button1: TButton;  Memo1: TMemo;  <> procedure  FormCreate(Sender: TObject);  <> procedure  RadioGroup1Click(Sender: TObject);	end; end.
type  TBitmapForm = class(TForm)  Chart1: TChart;  OpenDialog1: TOpenDialog;  Panel1: TPanel;  RadioGroup1: TRadioGroup;  BitBtn1: TBitBtn;  BitBtn3: TBitBtn;  CheckBox1: TCheckBox;  Series1: TBarSeries;  Button1: TButton;  Memo1: TMemo;  <> procedure  FormCreate(Sender: TObject);  <> procedure  RadioGroup1Click(Sender:  TObject);  <> procedure	end; end.  end.  ()()()()()()()()()()()()()()()()()()(
type  TBitmapForm = class(TForm)  Chart1: TChart;  OpenDialog1: TOpenDialog;  Panel1: TPanel;  RadioGroup1: TRadioGroup;  BitBtn1: TBitBtn;  CheckBox1: TCheckBox;  Series1: TBarSeries;  Button1: TButton;  Memo1: TMemo;  <> procedure  FormCreate(Sender: TObject);  <> procedure  RadioGroup1Click(Sender:  TObject);  <> procedure  BitBtn1Click(Sender: TObject);	end; end.
type  TBitmapForm = class(TForm)  Chart1: TChart;  OpenDialog1: TOpenDialog;  Panel1: TPanel;  RadioGroup1: TRadioGroup;  BitBtn1: TBitBtn;  BitBtn3: TBitBtn;  CheckBox1: TCheckBox;  Series1: TBarSeries;  Button1: TButton;  Memo1: TMemo;  <> procedure  FormCreate(Sender: TObject);  <> procedure  RadioGroup1Click(Sender: TObject);  <> procedure  BitBtn1Click(Sender: TObject);  <> procedure	end; end.
type  TBitmapForm = class(TForm)  Chart1: TChart;  OpenDialog1: TOpenDialog;  Panel1: TPanel;  RadioGroup1: TRadioGroup;  BitBtn1: TBitBtn;  BitBtn3: TBitBtn;  CheckBox1: TCheckBox;  Series1: TBarSeries;  Button1: TButton;  Memo1: TMemo;  <> procedure  FormCreate(Sender: TObject);  <> procedure  RadioGroup1Click(Sender: TObject);  <> procedure  BitBtn1Click(Sender: TObject);  <> procedure  CheckBox1Click(Sender: TObject);	end; end.
type  TBitmapForm = class(TForm)  Chart1: TChart;  OpenDialog1: TOpenDialog;  Panel1: TPanel;  RadioGroup1: TRadioGroup;  BitBtn1: TBitBtn;  BitBtn3: TBitBtn;  CheckBox1: TCheckBox;  Series1: TBarSeries;  Button1: TButton;  Memo1: TMemo;  <> procedure  FormCreate(Sender: TObject);  <> procedure  RadioGroup1Click(Sender: TObject);  <> procedure  BitBtn1Click(Sender: TObject);  <> procedure	end; end.

x1.Checked;	private
end;	{ Private declarations }
	public
<> procedure	{ Public declarations }
TBitmapForm.Button1Click(Sender	end;
: TObject);	,
begin	implementation
Chart1.BackImage:=nil;	•
end;	{\$R *.dfm}
	<> procedure
end.	TBitmapForm.FormCreate(Sender:
	TObject);
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	begin
	{ This is to show something random
7	in this example }
{***********	Series1.FillSampleValues(10);
*******	end;
{ TeeChart Delphi Component	,
Library }	<> procedure
{ Bubble Series Type Demo	TBitmapForm.RadioGroup1Click(S
}	ender: TObject);
{ Copyright (c) 1995-2001 by David	begin
Berneda }	· ·
{ All rights reserved }	Chart1.BackImageMode:=TTeeBac
{***********	kImageMode(RadioGroup1.ItemInd
*******	ex);
unit bubble;	end;
interface	<> procedure
	TBitmapForm.BitBtn1Click(Sender:
uses	TObject);
SysUtils, WinTypes, WinProcs,	begin
Messages, Classes, Graphics,	if OpenDialog1.Execute then
Controls,	begin
Forms, Dialogs, Chart, Series,	RadioGroup1.Enabled:=False;
ExtCtrls, StdCtrls, BubbleCh,	
Teengine, Buttons, TeeProcs;	Chart1.BackImage.LoadFromFile(O
	penDialog1.FileName);
type	RadioGroup1.Enabled:=True;
TBubbleForm = class(TForm)	end;
Chart1: TChart;	end;
Panel1: TPanel;	
CheckBox1: TCheckBox;	<> procedure
Timer1: TTimer;	TBitmapForm.CheckBox1Click(Sen
CheckBox2: TCheckBox;	der: TObject);
ComboBox1: TComboBox;	begin
Label1: TLabel;	
BitBtn3: TBitBtn;	Chart1.BackImageInside:=CheckBo

for t:=1 to 100 do	CheckBox3: TCheckBox;
BubbleSeries1.AddBubble(	BubbleSeries1: TBubbleSeries;
Date+t,	ZoomInButton: TSpeedButton;
Dute ty	ZoomOutButton: TSpeedButton;
Random(ChartSamplesMax), {	Memo1: TMemo;
< y value }	<> procedure
y value y	FormCreate(Sender: TObject);
ChartSamplesMax/(20+Random(25)	<> procedure
), { < radius value }	CheckBox1Click(Sender: TObject);
11	<> procedure
{ < label string }	Timer1Timer(Sender: TObject);
( last string)	<> procedure
GetDefaultColor(t)); { <	CheckBox2Click(Sender: TObject);
color }	<> procedure
end;	ComboBox1Change(Sender:
enu,	TObject);
<> procedure	<> procedure
TBubbleForm.CheckBox1Click(Sen	CheckBox3Click(Sender: TObject);
der: TObject);	function
begin	BubbleSeries1GetPointerStyle(Send
begin	er: TChartSeries;
Timer1.Enabled:=CheckBox1.Check	ValueIndex: Longint):
ed; { < on / off animation }	TSeriesPointerStyle;
end;	<> procedure
chu,	ZoomInButtonClick(Sender:
<> procedure	TObject);
TBubbleForm.Timer1Timer(Sender	<> procedure
: TObject);	ZoomOutButtonClick(Sender:
Var tmpColor:TColor;	TObject);
Begin	private
Timer1.Enabled:=False; { < stop	{ Private declarations }
the timer (this is optional) }	public
With BubbleSeries1 do	{ Public declarations }
Begin	
tmpColor:=ValueColor[0];	end;
1	implementation
Delete(0); { < remove the first	implementation
point }	(6D + 16)
{ Add a new random bubble }	{\$R *.dfm}
AddBubble(XValues.Last+1,	
{ < x value }	<> procedure
Dandom(ChaytSamplasMarr)	TBubbleForm.FormCreate(Sender:
Random(ChartSamplesMax), {	TObject);
< y value }	var t:Longint;
ChartCommissM-=/(20+D 1 (25)	begin
ChartSamplesMax/(20+Random(25)	
), { < radius value }	ComboBox1.ItemIndex:=Ord(psCirc
{ <	le); { < Circled Bubbles by default
label string }	}
tmpColor); {	BubbleSeries1.Clear;

ID 4 CI I I ( 1/1 / 00	
kBox2.Checked; { switch on/off	< color }
Marks }	end;
end;	if Random(100)<8 then
	begin
<> procedure	if
TBubbleForm.ComboBox1Change(S	ComboBox1.ItemIndex <combobox1< td=""></combobox1<>
ender: TObject);	.Items.Count-1 then
begin { the demo combobox1 allows	
changing Bubble style }	ComboBox1.ItemIndex:=ComboBox
	1.ItemIndex+1
BubbleSeries1.Pointer.Style:=TSerie	else
sPointerStyle(ComboBox1.ItemInde	ComboBox1.ItemIndex:=0;
x);	ComboBox1Change(Self);
end;	end;
	if (GetTickCount mod 1000)<=55
<> procedure	then
TBubbleForm.CheckBox3Click(Sen	begin
der: TObject);	with
begin	BubbleSeries1.GetHorizAxis.Title do
Chart1.Repaint;	if Angle>=90 then Angle:=Angle-
end;	90 else Angle:=270;
enu,	o eise ringie. 270,
function	with
TBubbleForm.BubbleSeries1GetPoi	BubbleSeries1.GetVertAxis.Title do
nterStyle(Sender: TChartSeries;	
ValueIndex: Longint):	if Angle>=90 then Angle:=Angle- 90 else Angle:=270;
TSeriesPointerStyle;	90 eise Aligie:-270;
begin	
if CheckBox3.Checked then	with BubbleSeries1.GetVertAxis
ii Checkbox3.Checkeu then	do
magulti-TS anias DaintanStyle (Danda	if LabelsAngle>=90 then
result:=TSeriesPointerStyle(Rando	LabelsAngle:=LabelsAngle-90 else
m(Ord(High(TSeriesPointerStyle))))	LabelsAngle:=270;
else	'a p ili c · 1 c ay · A ·
magult.=DubbleSowies1 Deinter Stude	with BubbleSeries1.GetHorizAxis
result:=BubbleSeries1.Pointer.Style;	do
end;	if LabelsAngle>=90 then
2 Samuel	LabelsAngle:=LabelsAngle-90 else
<> procedure	LabelsAngle:=270;
TBubbleForm.ZoomInButtonClick(	end;
Sender: TObject);	Timer1.Enabled:=True; { <
begin (117)	restart the timer }
Chart1.ZoomPercent(110);	end;
end;	
	<> procedure
<> procedure	TBubbleForm.CheckBox2Click(Sen
TBubbleForm.ZoomOutButtonClick	der: TObject);
(Sender: TObject);	begin
begin	
Chart1.ZoomPercent(90);	BubbleSeries1.Marks.Visible:=Chec

{\$R *.dfm}
Uses TeeAbout;
,
end.
9
\{\dagger \dagger \dag
**************************************
{ TeeChart }
{ TColoredForm Example
}
{ Copyright (c) 1995-2001 by David
Berneda }
{ All Rights Reserved }
{*********
*******
unit ucolor;
, , , , , , , , , , , , , , , , , , , ,
interface
uses
Wintypes, WinProcs, Messages,
SysUtils, Classes, Graphics,
Controls, Forms, Dialogs,
Teengine, Series, ExtCtrls, Chart,
StdCtrls, Buttons, teeprocs;
, , ,
type
TColoredForm = class(TForm)
Chart1: TChart;
Panel1: TPanel;
CheckBox1: TCheckBox;
BitBtn2: TBitBtn;
LineSeries1: TLineSeries;
PointSeries1: TPointSeries;
Memo1: TMemo;
<> procedure
FormCreate(Sender: TObject);
<> procedure
CheckBox1Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
implementation
mpromentum .
{\$R *.dfm}
{ <b>NR ^.aim</b> }

7
end;
end.
\\\\\\
8
{***********
\ ********
{ TeeChart Delphi Component
Library }
{ Chart Specs Form Demo
Chart Spees Form Demo
{ Copyright (c) 1995-2001 David
Berneda }
{ All rights reserved }
{***********************
\ ********
unit specs;
unit specs,
• 4 P
interface
USES  Craftile WinTrynes WinDress
SysUtils, WinTypes, WinProcs,
Messages, Classes, Graphics, Controls,
,
Forms, Dialogs, StdCtrls, Buttons,
ExtCtrls;
A
type
TChartSpecs = class(TForm)
TChartSpecs = class(TForm) Memo1: TMemo;
TChartSpecs = class(TForm)  Memo1: TMemo;  Panel2: TPanel;
TChartSpecs = class(TForm) Memo1: TMemo; Panel2: TPanel; Label1: TLabel;
TChartSpecs = class(TForm) Memo1: TMemo; Panel2: TPanel; Label1: TLabel; Image2: TImage;
TChartSpecs = class(TForm)  Memo1: TMemo; Panel2: TPanel; Label1: TLabel; Image2: TImage; BitBtn1: TBitBtn;
TChartSpecs = class(TForm) Memo1: TMemo; Panel2: TPanel; Label1: TLabel; Image2: TImage; BitBtn1: TBitBtn; Panel1: TPanel;
TChartSpecs = class(TForm)  Memo1: TMemo;  Panel2: TPanel;  Label1: TLabel;  Image2: TImage;  BitBtn1: TBitBtn;  Panel1: TPanel;  Memo2: TMemo;
TChartSpecs = class(TForm) Memo1: TMemo; Panel2: TPanel; Label1: TLabel; Image2: TImage; BitBtn1: TBitBtn; Panel1: TPanel; Memo2: TMemo; private
TChartSpecs = class(TForm)  Memo1: TMemo;  Panel2: TPanel;  Label1: TLabel;  Image2: TImage;  BitBtn1: TBitBtn;  Panel1: TPanel;  Memo2: TMemo;  private  { Private declarations }
TChartSpecs = class(TForm) Memo1: TMemo; Panel2: TPanel; Label1: TLabel; Image2: TImage; BitBtn1: TBitBtn; Panel1: TPanel; Memo2: TMemo; private
TChartSpecs = class(TForm)  Memo1: TMemo;  Panel2: TPanel;  Label1: TLabel;  Image2: TImage;  BitBtn1: TBitBtn;  Panel1: TPanel;  Memo2: TMemo;  private  { Private declarations }
TChartSpecs = class(TForm)  Memo1: TMemo;  Panel2: TPanel;  Label1: TLabel;  Image2: TImage;  BitBtn1: TBitBtn;  Panel1: TPanel;  Memo2: TMemo;  private  { Private declarations }  public
TChartSpecs = class(TForm)  Memo1: TMemo;  Panel2: TPanel;  Label1: TLabel;  Image2: TImage;  BitBtn1: TBitBtn;  Panel1: TPanel;  Memo2: TMemo;  private  { Private declarations }  public  { Public declarations }
TChartSpecs = class(TForm)  Memo1: TMemo;  Panel2: TPanel;  Label1: TLabel;  Image2: TImage;  BitBtn1: TBitBtn;  Panel1: TPanel;  Memo2: TMemo;  private  { Private declarations }  public  { Public declarations }

Berneda }	
{ All rights reserved }	<> procedure
{*************************************	TColoredForm.FormCreate(Sender:
*******	TObject);
unit ucrossh;	
	<> procedure
interface	AddColors(Series:TChartSeries);
	var step:Double;
uses	t:Longint;
SysUtils, WinTypes, WinProcs,	begin
Messages, Classes, Graphics,	With Series, GetVertAxis do
Controls,	begin
Forms, Dialogs, StdCtrls, Teengine,	step:=(Maximum-
Series, ExtCtrls, Chart, Buttons,	Minimum)/10.0;
TeeProcs;	for t:=0 to Count-1 do
,	
type	ValueColor[t]:=GetDefaultColor(
TCrossHairForm = class(TForm)	Trunc((YValue[t]-Minimum)/step) );
Chart1: TChart;	end;
LineSeries1: TLineSeries;	end;
Panel1: TPanel;	,
Label1: TLabel;	begin
BitBtn1: TBitBtn;	LineSeries1.FillSampleValues(100);
Label2: TLabel;	
CheckBox1: TCheckBox;	PointSeries1.FillSampleValues(100);
BitBtn2: TBitBtn;	Chart1.LeftAxis.AdjustMaxMin;
Memo1: TMemo;	AddColors(LineSeries1);
<> procedure	AddColors(PointSeries1);
FormCreate(Sender: TObject);	end;
<> procedure	
Chart1MouseMove(Sender:	<> procedure
TObject; Shift: TShiftState; X,	TColoredForm.CheckBox1Click(Sen
Y: Integer);	der: TObject);
<> procedure	begin
LineSeries1AfterDrawValues(Sende	
r: TObject);	Chart1.LeftAxis.Inverted:=CheckBo
<> procedure	x1.Checked;
CheckBox1Click(Sender: TObject);	end;
<> procedure	
BitBtn2Click(Sender: TObject);	
private	end.
{ Private declarations }	\\\\\\\\\\\
public	10
{ Public declarations }	{**********************
OldX,OldY:Longint;	*******
CrossHairColor:TColor;	{ TeeChart Delphi Component
CrossHairStyle:TPenStyle;	Library }
end;	{ Cross-Hair demo }
	{ Copyright (c) 1995-2001 by David

if (OldX<>-1) then	implementation
begin	Implementation
DrawCross(OldX,OldY); { draw	{\$R *.dfm}
old crosshair }	Uses udemutil;
OldX:=-1;	Oses ducinum,
end;	<> procedure
Circi,	TCrossHairForm.FormCreate(Send
{ check if mouse is inside Chart	er: TObject);
rectangle }	begin
if PtInRect( Chart1.ChartRect,	LineSeries1.FillSampleValues(30);
Point(X-	{ < some random values }
Chart1.Width3D,Y+Chart1.Height3	OldX:=-1; { initialize
D) ) then	variables }
begin	CrossHairColor:=clYellow;
DrawCross(x,y); { draw crosshair	CrossHairStyle:=psSolid;
at current position }	end;
{ store old position }	cnu,
OldX:=x;	< > procedure
OldY:=y;	<pre>&lt;&gt; procedure TCrossHairForm.Chart1MouseMov</pre>
{ set label text }	e(Sender: TObject; Shift:
With LineSeries1 do	TShiftState; X,
begin	Y: Integer);
GetCursorValues(tmpX,tmpY);	1. Integer),
	{ This <> procedure draws
{ < get values under mouse cursor }	the crosshair lines }
Label1.Caption:=GetVertAxis.Label	<> procedure
Value(tmpY)+	DrawCross(AX,AY:Integer);
' '+	begin
'	With Chart1, Canvas do
GetHorizAxis.LabelValue(tmpX);	begin
end;	Pen.Color:=CrossHairColor;
end;	Pen.Style:=CrossHairStyle;
end;	Pen.Mode:=pmXor;
ciiu,	Pen.Width:=1;
< > nwaaaduwa	,
<pre>&lt;&gt; procedure TCrossHairForm.LineSeries1AfterD</pre>	MoveTo(ax,ChartRect.Top-
rawValues(Sender: TObject);	Height3D);
begin	LineTo(ax,ChartRect.Bottom-
OldX:=-1; { Reset old mouse	Height3D);
position }	MayoTa(ChantDaat Laft   Width2Da
end;	MoveTo(ChartRect.Left+Width3D,a
cnu,	y);
< > procedure	LineTo(ChartRect.Right+Width3D,
<> procedure TCrossHoirForm Chack Box 1 Click (\$	`
TCrossHairForm.CheckBox1Click(S ender: TObject);	ay); end;
• /:	end;
if CheckBox1.Checked then	ciiu,
	Van tmnV tmnV Dankla
Chart1.Cursor:=crCross	Var tmpX,tmpY:Double;
else	begin

# <u>AC.&P</u>

BitBtn1: TBitBtn;	Chart1.Cursor:=crDefault;
CheckBox2: TCheckBox;	
PointSeries1: TPointSeries;	Chart1.OriginalCursor:=Chart1.Cu
FastLineSeries1: TFastLineSeries;	rsor;
DrawGrid: TCheckBox;	end;
Memo1: TMemo;	
<> procedure	<> procedure
FormCreate(Sender: TObject);	TCrossHairForm.BitBtn2Click(Send
<> procedure	er: TObject);
LineSeries1AfterDrawValues(Sende	begin
r: TObject);	
<> procedure	CrossHairColor:=EditColor(Self,Cr
CheckBox1Click(Sender: TObject);	ossHairColor);
<> procedure	end;
Timer1Timer(Sender: TObject);	
<> procedure	
CheckBox2Click(Sender: TObject);	end.
<> procedure	\\\\\\\\
DrawGridClick(Sender: TObject);	11
private	{********************
{ Private declarations }	*******
public	{ TeeChart Delphi Component
{ Public declarations }	Library }
XPercent, YPercent: Integer;	{ Custom Axis Drawing Demo
end;	}
	{ Copyright (c) 1995-2001 by David
implementation	Berneda }
	{ All rights reserved }
{\$R *.dfm}	{*********************
	*********
<> procedure	unit mulaxis;
TCustomAxisForm.FormCreate(Sen	
der: TObject);	interface
<> procedure	uses
CreateRandomPoints(Series:TChart	SysUtils, WinTypes, WinProcs,
Series);	Messages, Classes, Graphics,
var t,Old:Longint;	Controls,
begin	Forms, Dialogs, StdCtrls, ExtCtrls,
With Series do	Teengine, Chart, Series, Buttons,
begin	TeeProcs;
Clear;	
Old:=Longint(Random(1000));	type
for t:=1 to 100 do	TCustomAxisForm = class(TForm)
begin	Chart1: TChart;
Inc(Old,Longint(Random(20))-	LineSeries1: TLineSeries;
10);	Panel1: TPanel;
Add(Old,",clTeeColor);	CheckBox1: TCheckBox;
end;	Timer1: TTimer;

	end;
Pottom Avis Custom Draw (nosavis+1	,
BottomAxis.CustomDraw(posaxis+1	end;
0,posaxis+40,posaxis,DrawGrid.Che	1 .
cked);	begin
end;	XPercent:=50;
end;	YPercent:=50;
_	Randomize;
<> procedure	CreateRandomPoints(LineSeries1);
TCustomAxisForm.CheckBox1Click	
(Sender: TObject);	CreateRandomPoints(PointSeries1);
begin	
Chart1.AxisVisible:=not	CreateRandomPoints(FastLineSerie
CheckBox1.Checked;	s1);
	end;
Timer1.Enabled:=CheckBox1.Check	
ed;	<> procedure
end;	TCustomAxisForm.LineSeries1After
	DrawValues(Sender: TObject);
<> procedure	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
TCustomAxisForm.Timer1Timer(Se	var posaxis:longint;
` I	begin
nder: TObject);	With Chart1 do
begin	begin
if XPercent<95 then	{ Calculate axis position and
Inc(XPercent,5)	draw }
else XPercent:=5;	
if YPercent<97 then	PosAxis:=ChartRect.Left+Trunc(Ch
Inc(YPercent,3)	artWidth*YPercent/100.0);
else YPercent:=3;	LeftAxis.CustomDraw(posaxis-
Chart1.Repaint;	10,posaxis-
end;	40,posaxis,DrawGrid.Checked);
,	
<> procedure	
TCustomAxisForm.CheckBox2Click	PosAxis:=ChartRect.Left+Trunc(Ch
(Sender: TObject);	artWidth*(100.0-YPercent)/100.0);
begin	LeftAxis.CustomDraw(posaxis-
bv8iii	`*
Chart1.LeftAxis.Inverted:=CheckBo	10,posaxis-
x2.Checked;	40,posaxis,DrawGrid.Checked);
AZ.CHECKEU;	
Chart Datter A - I - I - Cl	D A CH (D (T )T
Chart1.BottomAxis.Inverted:=Chec	PosAxis:=ChartRect.Top+Trunc(Ch
kBox2.Checked;	artHeight*XPercent/100.0);
end;	
	BottomAxis.CustomDraw(posaxis+1
<> procedure	0,posaxis+40,posaxis,DrawGrid.Che
TCustomAxisForm.DrawGridClick(	cked);
Sender: TObject);	
begin	
Chart1.Repaint;	PosAxis:=ChartRect.Top+Trunc(Ch
end;	artHeight*(100.0-XPercent)/100.0);
7	

TObject);	
<> procedure	end.
CheckBox1Click(Sender: TObject);	
<> procedure	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
CheckBox2Click(Sender: TObject);	12
<> procedure	{*********************
Timer1Timer(Sender: TObject);	*******
<> procedure	{ TeeChart. THorizBarSeries
BitBtn3Click(Sender: TObject);	Data-Aware Demo }
private	{ Copyright (c) 1995-2001 by David
{ Private declarations }	Berneda }
public	{***********************
{ Public declarations }	********
DeltaZ:Integer;	unit udbhoriz;
<> procedure RecalcMarks;	,
end;	interface
implementation	{ This form shows Horizontal Bar
	Series attached to a Table }
{\$R *.dfm}	
(wit idin)	uses
<> procedure	SysUtils, WinTypes, WinProcs,
TDBHorizBarForm.FormShow(Sen	Messages, Classes, Graphics,
der: TObject);	Controls,
begin	Forms, Dialogs, DB, DBTables,
DeltaZ:=-5; { < this variable is	Chart, Series, ExtCtrls, DbChart,
used for animation only }	StdCtrls,
used for animation only j	Teengine, Buttons, teeprocs;
{ Here we set the Series order }	rengine, Buttons, teeproes,
With DBChart1.SeriesList do	type
Begin	TDBHorizBarForm = class(TForm)
Items[0]:=HorizBarSeries2;	Table1: TTable;
Items[1]:=HorizBarSeries2;	DBChart1: TDBChart;
End;	Panel1: TPanel;
,	CheckBox1: TCheckBox;
end;	RadioGroup1: TRadioGroup;
/ > procedure	Timer1: TTimer;
<pre>&lt;&gt; procedure TDBHorizBarForm.RecalcMarks;</pre>	BitBtn2: TBitBtn;
	BitBtn3: TBitBtn;
Begin ( Hara the Sories Marks and	HorizBarSeries1:
{ Here the Series Marks and associated Axis are	THorizBarSeries:
	HorizBarSeries; HorizBarSeries2:
adjusted depending on MultiBar (Stacked or not )	THorizBarSeries;
7	CheckBox2: TCheckBox;
if	,
	Memo1: TMemo;
(HorizBarSeries1.MultiBar=mbStac	<> procedure FormShow(Sondow TObicot):
ked) or	FormShow(Sender: TObject);
(HavinDayCavin-1 MaleD	<> procedure
(HorizBarSeries1.MultiBar=mbStac	RadioGroup1Click(Sender:

ecked; { < 3 dimensions }	ked100) then
end;	Begin
enu,	Degin
< > nuocoduro	DBChart1.Series[0].Marks.Visible:=
<> procedure TDBHorizBarForm.CheckBox2Clic	False;
	raise,
k(Sender: TObject);	DDChautt Savias[1] Maults Visibles—
begin	DBChart1.Series[1].Marks.Visible:= True;
Timer1.Enabled:=CheckBox2.Check	True,
ed; { < start / stop animation }	HorizBarSeries1.HorizAxis:=aTopA
end;	xis;
enu,	A13,
/ > nmaaaduma	HorizBarSeries2.HorizAxis:=aTopA
<pre>&lt;&gt; procedure TDBHorizBarForm.Timer1Timer(S</pre>	xis;
	end
ender: TObject); var tmp:Integer;	else
	Begin
begin Timori Enabled:-Felse	Degin
Timer1.Enabled:=False;	DBChart1.Series[0].Marks.Visible:=
{ here is the animation code }	True;
two - DDCh aut1 Chaut2DDaysaut1D	True,
tmp:=DBChart1.Chart3DPercent+D	DBChart1.Series[1].Marks.Visible:=
eltaZ;	True;
if tmp<3 then	True,
Begin	HorizBarSeries1.HorizAxis:=aBotto
DeltaZ:=5;	mAxis;
tmp:=3;	maxis,
end;	HorizBarSeries2.HorizAxis:=aTopA
if tmp>90 then	xis;
begin	end
DeltaZ:=-5;	end;
tmp:=90;	Citu,
end;	<> procedure
DBChart1.Chart3DPercent:=tmp;	TDBHorizBarForm.RadioGroup1Cl
DBChart1.Legend.TopPos:=tmp;	ick(Sender: TObject);
if Random(100)=1 then	begin
	{ Change the Series Stacked }
HorizBarSeries1.BarStyle:=TBarSty	( Change the Berres Blackett )
le(Random(Ord(High(TBarStyle))));	HorizBarSeries1.MultiBar:=TMulti
if Random(100)=1 then	Bar(RadioGroup1.ItemIndex);
	RecalcMarks;
HorizBarSeries2.BarStyle:=TBarSty	end;
le(Random(Ord(High(TBarStyle))));	ciiu,
if Random(100)=1 then	< > nroanduro
	<pre>&lt;&gt; procedure TDBHorizBarForm.CheckBox1Clic</pre>
RadioGroup1.ItemIndex:=Random(	k(Sender: TObject);
RadioGroup1.Items.Count);	• • • • • • • • • • • • • • • • • • • •
Timer1.Enabled:=True;	begin
end;	DDChart1 Viaw2Di-Charl Day1 Ch
	DBChart1.View3D:=CheckBox1.Ch

_
Chart1: TChart;
Series1: TLineSeries;
Series2: TLineSeries;
Series3: TLineSeries;
Series4: TLineSeries;
Chart2: TChart;
Series5: TBarSeries;
Series6: TBarSeries;
Series7: TBarSeries;
Panel2: TPanel;
,
Chart3: TChart;
Series8: THorizBarSeries;
Series9: THorizBarSeries;
Series10: THorizBarSeries;
Chart4: TChart;
Series11: TAreaSeries;
Series12: TAreaSeries;
Chart5: TChart;
Series13: TPointSeries;
Series14: TPointSeries;
Series15: TPointSeries;
Series16: TPointSeries;
Series17: TPointSeries;
Series18: TPointSeries;
Chart6: TChart;
Series19: TPieSeries;
Chart7: TChart;
Series20: TFastLineSeries;
Series21: TFastLineSeries;
Series22: TFastLineSeries;
Series23: TFastLineSeries;
Series24: TFastLineSeries;
Chart8: TChart;
Series26: TArrowSeries;
Series27: TArrowSeries;
Chart9: TChart;
Series28: TBubbleSeries;
Chart10: TChart;
Series29: TGanttSeries;
Chart11: TChart;
Series30: TChartShape;
Series31: TChartShape;
Series32: TChartShape;
Button2: TButton;
Panel3: TPanel;
ListBox1: TListBox;
CheckBox1: TCheckBox;
CheckBox2: TCheckBox;

<> procedure
TDBHorizBarForm.BitBtn3Click(Se
nder: TObject);
var tmp:TChartSeries;
begin
U
{ This code "swaps" the two series }
With DBChart1.SeriesList do
Begin
tmp:=Items[0];
Items[0]:=Items[1];
Items[1]:=tmp;
End;
RecalcMarks;
DBChart1.Repaint;
end:
CHU,
end.
\\\\\\\\\\\
13
{***********
\ *******
\$
{ TeeChart Delphi Component
Library v3 }
{ Copyright (c) 1995-2001 by David
Berneda }
{ All rights reserved }
{***********************
******
unit teebasic;
{\$P-} { < Delphi 1.0 compatibility }
(*- ) ( 2 sipin 210 companionity)
interface
intellace
uses
SysUtils, WinTypes, WinProcs,
Messages, Classes, Graphics,
Controls,
Forms, Dialogs, ExtCtrls, TeeProcs,
TeEngine, Chart, StdCtrls, Series,
TeeShape, GanttCh, BubbleCh,
ArrowCha, Buttons;
Till on enay buttons,
tyna
TDomo Form - class(TForm)
TDemoForm = class(TForm)
Notebook1: TNotebook;

<> procedure	Image1: TImage;
SpeedButton2Click(Sender:	Timer1: TTimer;
TObject);	ScrollBar1: TScrollBar;
<> procedure	Label1: TLabel;
SpeedButton3Click(Sender:	BitBtn1: TBitBtn;
TObject);	BitBtn2: TBitBtn;
<> procedure	SpeedButton1: TSpeedButton;
CheckBox3Click(Sender: TObject);	SpeedButton2: TSpeedButton;
<> procedure	SpeedButton3: TSpeedButton;
ComboBox1Change(Sender:	SpeedButton4: TSpeedButton;
TObject);	Label2: TLabel;
<> procedure	Label3: TLabel;
Image1Click(Sender: TObject);	,
<> procedure	CheckBox3: TCheckBox;
ScrollBar2Change(Sender:	ComboBox1: TComboBox;
TObject);	ComboBox2: TComboBox;
<> procedure	ScrollBar2: TScrollBar;
Series28GetMarkText(Sender:	Button3: TButton;
TChartSeries;	CheckBox4: TCheckBox;
ValueIndex: Integer; var	Label4: TLabel;
<u> </u>	Label5: TLabel;
MarkText: string);	Label6: TLabel;
<> procedure	Label7: TLabel;
Button3Click(Sender: TObject);	ScrollBar3: TScrollBar;
<> procedure	Label8: TLabel;
CheckBox4Click(Sender: TObject);	<> procedure
<> procedure	ListBox1Click(Sender: TObject);
FormShow(Sender: TObject);	<> procedure
<> procedure	FormCreate(Sender: TObject);
ScrollBar3Change(Sender:	<> procedure
TObject);	CheckBox1Click(Sender: TObject);
private	<> procedure
{ Private declarations }	CheckBox2Click(Sender: TObject);
public	<> procedure
{ Public declarations }	ScrollBar1Change(Sender:
tmpSeries,	TObject);
tmpIndex,	<> procedure
tmpRandom,	Timer1Timer(Sender: TObject);
tmpTimes:Longint;	` *
Function TheChart: TChart;	<> procedure Putton2Click(Sondow TObicat):
<> procedure	Button2Click(Sender: TObject);
HorizScroll(Const Percent:Double);	<> procedure
<> procedure	BitBtn1Click(Sender: TObject);
VertScroll(Const Percent:Double);	<> procedure
<> procedure	BitBtn2Click(Sender: TObject);
ScrollAxis(Axis:TChartAxis; Const	<> procedure
Percent:Double);	SpeedButton1Click(Sender:
end;	TObject);
CHU,	<> procedure
:	SpeedButton4Click(Sender:
implementation	TObject);

Comba Dag Mana Ladan A	
ComboBox2.ItemIndex:=0;	(OD + 10 )
ListBox1.ItemIndex :=0;	{\$R *.dfm}
ListBox1Click(Self);	Uses TeeAbout;
end;	
	<> procedure
<> procedure	TDemoForm.ListBox1Click(Sender:
TDemoForm.CheckBox1Click(Send	TObject);
er: TObject);	var t:Longint;
begin	begin
TheChart.View3D:=CheckBox1.Che	NoteBook1.PageIndex:=ListBox1.Ite
cked;	mIndex;
	With TheChart do
ScrollBar2.Enabled:=TheChart.Vie	begin
w3D;	tmpTimes:=-1;
	AnimatedZoom:=True;
ScrollBar3.Enabled:=TheChart.Vie	AnimatedZoomSteps:=4;
w3D;	for t:=0 to SeriesCount-1 do
end;	With Series[t] do
<> procedure	FillSampleValues(NumSampleValue
TDemoForm.CheckBox2Click(Send	s);
er: TObject);	UndoZoom;
begin	CheckBox1.Checked :=View3D;
	ScrollBar2.Enabled :=View3D;
Timer1.Enabled:=CheckBox2.Check	Seronbur 21211usieu 1 + 1e (1e 2)
ed;	ScrollBar2.Position:=Chart3DPerce
,	nt;
ScrollBar1.Enabled:=Timer1.Enable	if Series[0] is TPieSeries then
d;	
end;	ScrollBar3.Position:=View3DOption
	s.Elevation
<> procedure	else
TDemoForm.ScrollBar1Change(Sen	
der: TObject);	ScrollBar3.Position:=View3DOption
begin	s.Rotation;
	ScrollBar3.Enabled:=View3D;
Timer1.Interval:=ScrollBar1.Positio	end;
n;	end;
end;	Citu,
,	<> procedure
Function	TDemoForm.FormCreate(Sender:
TDemoform.TheChart:TChart;	TObject);
begin	begin
with NoteBook1 do	tmpTimes:=-1;
result:=(Pages.Objects[PageIndex]	unp i mes1,
as TPage).Controls[0] as TChart;	ScrollBar1.Position:=Timer1.Interva
end;	
ciu,	l;
	ComboBox1.ItemIndex:=0;

begin
RadiusValues[tmpIndex]:=tmpX;
if tmpRandom>0 then
tmpX:=XValues[tmpIndex]+1 else
tmpX:=XValues[tmpIndex]-1;
XValues[tmpIndex]:=tmpX;
if Random(10)<5 then
tmpX:=YValues[tmpIndex]+50
else
tmpX:=YValues[tmpIndex]-50;
YValues[tmpIndex]:=tmpX;
end
else tmpTimes:=0;
end
else
With
Series[tmpSeries].MandatoryValueL ist do
Value[tmpIndex]:=Value[tmpIndex] +tmpRandom;
Dec(tmpTimes);
Repaint;
end;
end;
5: (TheChart[0] as
TPieSeries).Rotate(358);
end;
end;
<> procedure
TDemoForm.Button2Click(Sender: TObject);
begin
Close;
end;
Citu,
<> procedure
TDemoForm.BitBtn1Click(Sender:
TObject);
begin
TheChart.ZoomPercent(120);
Button3.Enabled:=True;
end;

<> procedure
TDemoForm.Timer1Timer(Sender:
TObject);
var t:Longint;
tmpX:Double;
begin
With NoteBook1 do
Case PageIndex of
0,3,4,6: With TheChart do
begin
for t:=0 to SeriesCount-1
do
With Series[t] do
begin
tmpX:=XValues[1]-
XValues[0];
Delete(0);
AddXY(
XValues.Last+tmpX,
21 valuesi Lust · tilipzi,
YValues.Last+Random(100)-
50,",clTeeColor);
end;
end;
1,2,8: With TheChart do
begin
if (tmpTimes=-1) then
begin
begin
tmpSeries:=Random(SeriesCount);
tmpIndex
-
:=Random(Series[tmpSeries].Count)
tmpTimes
:=Random(10);
Kandom(10),
tmnDandamı=2*Daund(Dandam(50
tmpRandom:=2*Round(Random(50
)-25.0);
end;
if tmpSeries<>-1 then
begin
if PageIndex=8 then
With (Series[tmpSeries]
as TBubbleSeries) do
begin
tmpX:=RadiusValues[tmpIndex]+tm
pRandom;
if tmpX>=2 then

TDomoForm CroadDutton1Click(Co	∠ > muo ao duumo
TDemoForm.SpeedButton1Click(Se	<pre>&lt;&gt; procedure TD Frame PitPt-2CH (Condense)</pre>
nder: TObject);	TDemoForm.BitBtn2Click(Sender:
begin	TObject);
HorizScroll(10);	begin
end;	TheChart.ZoomPercent(80);
	Button3.Enabled:=True;
<> procedure	end;
TDemoForm.SpeedButton4Click(Se	
nder: TObject);	<> procedure
begin	TDemoForm.ScrollAxis(Axis:TChar
HorizScroll(-10);	tAxis; Const Percent:Double);
end;	var Amount:Double;
chu,	begin
<> procedure	With Axis do
TDemoForm.SpeedButton2Click(Se	begin
nder: TObject);	Amount:=-((Maximum-
begin	Minimum)/(100.0/Percent));
VertScroll(-10);	SetMinMax(Minimum-
end;	Amount, Maximum-Amount);
	end;
<> procedure	end;
TDemoForm.SpeedButton3Click(Se	
nder: TObject);	<> procedure
begin	TDemoForm.HorizScroll(Const
VertScroll(10);	Percent:Double);
end;	begin
chu,	begin
z > wwo.coduwo	ScrollAxis(TheChart.TopAxis,Perce
<pre>&lt;&gt; procedure TDomo Form Chash Pay 2 Click (Sand)</pre>	•
TDemoForm.CheckBox3Click(Send	nt);
er: TObject);	
begin	ScrollAxis(TheChart.BottomAxis,Pe
(TheChart[0] as	rcent);
TPieSeries).Circled:=CheckBox3.Ch	Button3.Enabled:=True;
ecked;	end;
end;	
	<> procedure
<> procedure	TDemoForm.VertScroll(Const
TDemoForm.ComboBox1Change(Se	Percent:Double);
nder: TObject);	begin
begin	9
With (TheChart[0] as	ScrollAxis(TheChart.LeftAxis,Perce
TCustomBarSeries) do	nt);
MultiBar:=TMultiBar((Sender as	1179
	ScrollAxis(TheChart.RightAxis,Perc
TComboBox).ItemIndex);	`
end;	ent);
	Button3.Enabled:=True;
<> procedure	end;
TDemoForm.Image1Click(Sender:	
TObject);	<> procedure
<del></del>	

	[ ·
Series11.Stairs:=CheckBox4.Checke	begin
d;	Timer1.Enabled:=False;
	With TTeeAboutForm.Create(Self)
Series12.Stairs:=CheckBox4.Checke	do
d;	try
end;	ShowModal;
	finally
<> procedure	Free;
TDemoForm.FormShow(Sender:	end;
TObject);	Timer1.Enabled:=True;
begin	end;
Timer1.Enabled:=True; { < start	chu,
animation }	<> procedure
end;	TDemoForm.ScrollBar2Change(Sen
Viin,	
<> procedure	der: TObject);
TDemoForm.ScrollBar3Change(Sen	begin
	THE CLASSIC AND ASSESSED.
der: TObject);	TheChart.Chart3DPercent:=ScrollB
begin	ar2.Position;
if ScrollBar3.Enabled then	end;
With TheChart.View3DOptions do	
begin	<> procedure
Orthogonal:=False;	TDemoForm.Series28GetMarkText(
if TheChart[0] is TPieSeries then	Sender: TChartSeries;
Elevation:=ScrollBar3.Position	ValueIndex: Integer; var
else	MarkText: string);
Rotation:=ScrollBar3.Position;	begin
end;	if ValueIndex=3 then
end;	MarkText:='USA' else
	if ValueIndex=5 then
end.	MarkText:='UK' else
\\\\\\\	if ValueIndex=7 then
14	MarkText:='Germany' else
	MarkText:='';
{*********	end;
\ ********	,
{ TeeChart Delphi Component	<> procedure
Library }	TDemoForm.Button3Click(Sender:
{ Digital Series and Legend Last	TObject);
Values Demo }	begin
,	TheChart.UndoZoom;
{ Copyright (c) 1995-2001 by David Berneda }	·
,	Button3.Enabled:=False;
{ All rights reserved }	end;
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
*******	<> procedure
{\$P-} { < VERY IMPORTANT	TDemoForm.CheckBox4Click(Send
*************  {\$P-} { < VERY IMPORTANT WHEN USING OnGet EVENTS }	TDemoForm.CheckBox4Click(Send er: TObject);
\{\frac{\circ}{***************\}}{\{\frac{\circ}{\circ}\} \{\left\{ < VERY IMPORTANT}}	TDemoForm.CheckBox4Click(Send

Index: Longint; var LegendText:	interface
String);	{ This forms show 4 line series in
<> procedure	"Stairs" mode.
CheckBox4Click(Sender: TObject);	Stairs mode.
• • • • • • • • • • • • • • • • • • • •	(LingConingleAring Towns )
private	(LineSeries1.Stairs := True;)
{ Private declarations }	
public	Legend Style is "lsLastValues"
{ Public declarations }	meaning the Legend will draw the
end;	Last value for each Series.
	}
implementation	
	uses
{\$R *.dfm}	SysUtils, WinTypes, WinProcs,
	Messages, Classes, Graphics,
<> procedure	Controls,
TDigitalForm.FormCreate(Sender:	Forms, Dialogs, Chart, Series,
TObject);	ExtCtrls, StdCtrls, Teengine,
var t,tt:Longint;	Buttons,
begin	TeeProcs;
~~g	
Chart1.ApplyZOrder:=CheckBox4.	type
Checked; { ZOrder or not	TDigitalForm = class(TForm)
ZOrder }	Chart1: TChart;
Chart1.Legend.Inverted:=True;	,
	LineSeries1: TLineSeries;
{ Fill Series with random values }	LineSeries2: TLineSeries;
for t:=0 to Chart1.SeriesCount-1 do	LineSeries3: TLineSeries;
With Chart1.Series[t] do	LineSeries4: TLineSeries;
begin	Panel1: TPanel;
Clear;	CheckBox1: TCheckBox;
for tt:=1 to 100 do Add(	Timer1: TTimer;
2*t+Random(2), ", clTeeColor);	CheckBox2: TCheckBox;
end;	CheckBox3: TCheckBox;
end;	BitBtn3: TBitBtn;
	CheckBox4: TCheckBox;
<> procedure	Memo1: TMemo;
TDigitalForm.CheckBox1Click(Send	<> procedure
er: TObject);	FormCreate(Sender: TObject);
begin	<> procedure
	CheckBox1Click(Sender: TObject);
Timer1.Enabled:=CheckBox1.Check	<> procedure
ed; { start / stop animation }	Timer1Timer(Sender: TObject);
end;	<> procedure
	CheckBox2Click(Sender: TObject);
<> procedure	<> procedure
TDigitalForm.Timer1Timer(Sender:	CheckBox3Click(Sender: TObject);
TObject);	<> procedure
var t:Longint;	Chart1GetLegendText(Sender:
begin	TCustomAxisPanel;
Timer1.Enabled:=False; { < stop	,
Timei I. Enableu: - Faise; { \ stop	LegendStyle: TLegendStyle;

t(Sender: TCustomAxisPanel;	the timer }
LegendStyle: TLegendStyle; Index:	,
Longint; var LegendText: String);	{ Now, add a new point to each
begin	Series }
{ we want to show the Series Title	for t:=0 to Chart1.SeriesCount-1 do
as well as the Last Values }	With Chart1.Series[t] do Add(
if LegendStyle=lsLastValues then	2*t+Random(2),'',clTeeColor);
LegendText:=LegendText+'>	
'+Chart1.Series[Index].Title;	{ Scroll the Horizontal Axis }
end;	With Chart1.BottomAxis do { <
	with the Horizontal Axis }
<> procedure	Begin
TDigitalForm.CheckBox4Click(Send	Automatic := False; { < we
er: TObject);	dont want automatic scaling }
begin	Maximum :=
	LineSeries1.XValues.Last;
Chart1.ApplyZOrder:=CheckBox4.	Minimum := Maximum - 100; {
Checked;	we want to see the last 100 points
Chart1.Repaint;	only }
end;	End;
	{ re-start timer }
	Timer1.Enabled:=True;
end.	end;
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	,
15	<> procedure
<b>{**********</b>	TDigitalForm.CheckBox2Click(Send
*******	er: TObject);
{ TeeChart. Draw Example	begin
}	
{ Copyright (c) 1995-2001 by David	Chart1.View3D:=CheckBox2.Check
Berneda }	ed;
{ All Rights Reserved }	end;
<b>{*****************</b>	
*******	<> procedure
unit udraw;	TDigitalForm.CheckBox3Click(Send
	er: TObject);
interface	begin
	if CheckBox3.Checked then
{ THIS EXAMPLE SHOWS HOW	
TO DRAW ADDITIONAL	Chart1.Legend.LegendStyle:=lsLast
CUSTOMIZED THINGS TO A	Values
CHART COMPONENT }	else
uses	
SysUtils, WinTypes, WinProcs,	Chart1.Legend.LegendStyle:=lsAuto
Messages, Classes, Graphics,	;
Controls,	end;
Forms, Dialogs, Chart, Series,	
ExtCtrls, Teengine, StdCtrls,	<> procedure
Buttons,	TDigitalForm.Chart1GetLegendTex

_
<> procedure
TDrawForm.LineSeries1BeforeDra
wValues(Sender: TObject);
Const
MyColors:array[15] of TColor=
(clNavy,
clGreen,
clYellow,
clRed,
\$0000080 { very red }
);
var t,partial:Longint;
tmpRect:TRect;
YPosition:Longint;
tmpYCenterValue:Double;
begin
With Chart1 do
Begin
{ we will divide the total chart
width by 5 }
tmpRect:=ChartRect;
tmpRect.Right:=tmpRect.Left;
partial:=ChartWidth div 5;
partial.—Chart Width div 3,
{ change the brush style }
Canvas.Brush.Style:=bsDiagCross;
Canvas.Pen.Style:=psClear;
Canvas.i en.stylepsclear,
(for each section fill with a
{ for each section, fill with a
specific color }
for t:=1 to 5 do
Begin
{ adjust the rectangle dimension
}
tmpRect.Right
:=tmpRect.Right+partial+1;
(
{ set the brush color }
Canvas.Brush.Color:=MyColors[t];
6 . 4111.)
{ paint !!! }
With tmpRect do
Canvas.Rectangle(
Left+Width3D,Top-
Height3D,Right+Width3D,Bottom-
Height3D);

T
TeeProcs;
,
type
TDrawForm = class(TForm)
Chart1: TChart;
LineSeries1: TLineSeries;
Panel1: TPanel;
BitBtn3: TBitBtn;
CheckBox1: TCheckBox;
Timer1: TTimer;
BitBtn1: TBitBtn;
CheckBox2: TCheckBox;
Memo1: TMemo;
<> procedure
FormCreate(Sender: TObject);
<> procedure
LineSeries1BeforeDrawValues(Send
er: TObject);
<> procedure
LineSeries1AfterDrawValues(Sende
r: TObject);
<> procedure
Timer1Timer(Sender: TObject);
<> procedure
CheckBox1Click(Sender: TObject);
<> procedure
BitBtn1Click(Sender: TObject);
<> procedure
CheckBox2Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
Percent:Double;
DeltaPercent:Integer;
end;
,
implementation
{\$R *.dfm}
(* ******)
<> procedure
TDrawForm.FormCreate(Sender:
TObject);
begin
Percent:=50; { < used for this
demo only }
LineSeries1.FillSampleValues(20);
end;
ciiu,

ent*(MaxValue-MinValue)/100.0;	
che (man e mae man e mae), 10010;	{ adjust rectangle }
{ then calculate the Screen Pixel	tmpRect.Left:=tmpRect.Right;
coordinate of the above value }	end;
esorumuse or the upove value ;	ciru,
YPosition:=LeftAxis.CalcYPosValue	{ first calculate the middle vertical
(tmpYCenterValue);	value (based on LineSeries points) }
(	With LineSeries1.YValues do
{ change pen and draw the line }	With Emeseriest.1 values do
Pen.Width:=3;	tmpYCenterValue:=MinValue+Perc
Pen.Style:=psSolid;	ent*(MaxValue-MinValue)/100.0;
Pen.Color:=clRed;	
Tenicolori enteu;	{ then calculate the Screen Pixel
MoveTo(ChartRect.Left,YPosition);	coordinate of the above value }
Wiever of Chartereet. Eet g. 11 ostilon);	coordinate of the above value
LineTo(ChartRect.Right,YPosition);	YPosition:=LeftAxis.CalcYPosValue
Emero(Chartiteetinght, 11 osition),	(tmpYCenterValue);
LineTo(ChartRect.Right+Width3D,	(timp i center varae);
YPosition-Height3D);	With Canvas do
Trostron freigness;	begin
{ change font and draw some text	{ change pen and draw the line }
above the line }	Pen.Width:=3;
above the me ;	Pen.Style:=psSolid;
Font.Name:='Arial';	Pen.Color:=clRed;
Tonch value. Titlat ;	Ten.color. checu,
{ VERY IMPORTANT !!!!!! }	MoveTo(ChartRect.Left,YPosition);
{ THIS IS NECESSARY IF	move to (Chartreettett, 11 ostion),
YOU'RE GOING TO PRINT !!!! }	LineTo(ChartRect.Left+Width3D,Y
{ IT MAKES FONT SIZES TO	Position-Height3D);
WORK FINE BOTH AT SCREEN	rosition freighte 2),
AND PRINTER. }	LineTo(ChartRect.Right+Width3D,
,	YPosition-Height3D);
Font.Height:=-24; { < express	end;
font size in "Height", NOT "Size" }	end;
, , , , , , , , , , , , , , , , , , , ,	end;
Font.Color:=clYellow;	CITA',
Font.Style:=[fsBold];	<> procedure
[22202]	TDrawForm.LineSeries1AfterDraw
{ Set transparent background }	Values(Sender: TObject);
Brush.Style:=bsClear;	Var YPosition:Longint;
Di udition jier botteni j	tmpYCenterValue:Double;
{ Output some text }	begin
TextOut( ChartRect.Left+20,	With Chart1, Canvas do
YPosition-24,	Begin
'This is	{ first calculate the middle vertical
'+FloatToStr(tmpYCenterValue));	value (based on LineSeries points) }
end;	With LineSeries1.YValues do
end;	THE DIMENSION THEORY
Chu,	tmpYCenterValue:=MinValue+Perc
	mp i contai i uiuci i iiiii i uiuci i ci c

111111111111111111111111111111111111111	<> procedure
16	TDrawForm.Timer1Timer(Sender:
	TObject);
{*************************************	begin
*******	if Percent+DeltaPercent>100 then
{ TeeChart. TChart Component	begin
}	Percent:=100;
{ Copyright (c) 1995-2001 by David	DeltaPercent:=-5;
Berneda }	end
{ All Rights Reserved }	else
{***********	if Percent+DeltaPercent<0 then
*******	begin
unit ufast;	Percent:=0;
	DeltaPercent:=5;
interface	end
	else
uses	Percent:=Percent+DeltaPercent;
SysUtils, WinTypes, WinProcs,	Chart1.Repaint;
Messages, Classes, Graphics,	end;
Controls,	
Forms, Dialogs, StdCtrls, ExtCtrls,	<> procedure
Teengine, Chart, Series, Buttons,	TDrawForm.CheckBox1Click(Sende
TeeProcs;	r: TObject);
	begin
type	oeg.n
TFastLineForm = class(TForm)	Timer1.Enabled:=CheckBox1.Check
Chart1: TChart;	ed;
Panel1: TPanel;	DeltaPercent:=5;
Button1: TButton;	end;
CheckBox1: TCheckBox;	
CheckBox2: TCheckBox;	<> procedure
FastLineSeries1: TFastLineSeries;	TDrawForm.BitBtn1Click(Sender:
FastLineSeries2: TFastLineSeries;	TObject);
CheckBox3: TCheckBox;	begin
BitBtn3: TBitBtn;	{ try with and without this>
Memo1: TMemo;	Chart1.PrintResolution := -100; }
<> procedure	Chart1.PrintLandscape;
FormCreate(Sender: TObject);	end;
<> procedure	,
Button1Click(Sender: TObject);	<> procedure
<> procedure	TDrawForm.CheckBox2Click(Sende
CheckBox2Click(Sender: TObject);	r: TObject);
<> procedure	begin
CheckBox1Click(Sender: TObject);	
<> procedure	Chart1.View3D:=CheckBox2.Check
CheckBox3Click(Sender: TObject);	ed;
private	end;
{ Private declarations }	
public	end.

## <u>AC.&P</u>

Chart1.AnimatedZoom:=False;	{ Public declarations }
t1:=GetTickCount;	end;
for t:=1 to 30 do	
Chart1.ZoomPercent(105); { 5%	implementation
zoom in }	
for t:=1 to 30 do	{\$R *.dfm}
Chart1.ZoomPercent(95); { 5%	
zoom out }	<> procedure
t2:=GetTickCount;	TFastLineForm.FormCreate(Sender
Chart1.AnimatedZoom:=True;	: TObject);
Chart1.UndoZoom;	var t,tmpRandom:Longint;
finally	begin
Screen.Cursor:=crDefault;	{ This can speed up things a little
end;	}
Showmessage('Time to plot 2000	TeeEraseBack:=False;
points'+#13+	,
'61 times: '+#13+	{ Make the chart flicker by default,
IntToStr(t2-t1)+'	only for this demo }
milliseconds.');	Chart1.BufferedDisplay:=False;
end;	
	Chart1.View3d:=False;
<> procedure	Chart1.Legend.Visible:=False;
TFastLineForm.CheckBox2Click(Se	Chart1.Title.Visible:=False;
nder: TObject);	Chart1.Foot.Visible:=False;
begin	Chartin out visible. Taise,
	{ some speed improvement if }
Chart1.AxisVisible:=CheckBox2.Ch	TeeDefaultCapacity:=2000;
ecked;	1 ccDclauticapacity:-2000,
end;	{ 2000 random points }
	Randomize;
<> procedure	for t:=1 to 1000 do
TFastLineForm.CheckBox1Click(Se	
nder: TObject);	begin
begin	tmpRandom:=Random(Abs(500-
{ Setting this to False speeds up	t))-(Abs(500-t) div 2);
drawing	FastLineSeries1.Add(1000-
with maximized charts on some	t+tmpRandom,",clTeeColor);
video drivers }	Fact in Sovies 2 Add(t   town Dondons
video dilivers j	FastLineSeries2.Add(t+tmpRandom,
Chart1.BufferedDisplay:=CheckBox	",clTeeColor);
1.Checked;	end;
end;	end;
chu,	
<> procedure	<pre>&lt;&gt; procedure TFtLingForm Poster 165 - 166 - 16</pre>
TFastLineForm.CheckBox3Click(Se	TFastLineForm.Button1Click(Sende
nder: TObject);	r: TObject);
begin	var t1,t2,t:Longint;
Degin	begin
Chart1.ClipPoints:=CheckBox3.Che	Screen.Cursor:=crHourGlass;
Charti. Chpi viitis.—Checkduxs. Che	try

BitBtn21: TBitBtn;
Panel2: TPanel;
Label21: TLabel;
Label22: TLabel;
Label23: TLabel;
Label24: TLabel;
Label25: TLabel;
Label26: TLabel;
Label27: TLabel;
Label28: TLabel;
Label29: TLabel;
Label30: TLabel;
Label31: TLabel;
Label32: TLabel;
Label32: TLabel;
Label34: TLabel;
Label35: TLabel;
Label36: TLabel;
,
Label37: TLabel; Label38: TLabel;
,
Label39: TLabel;
Label40: TLabel;
BitBtn22: TBitBtn;
Label14: TLabel;
Panel1: TPanel;
<> procedure
BitBtn3Click(Sender: TObject);
<> procedure
BitBtn5Click(Sender: TObject);
<> procedure
BitBtn6Click(Sender: TObject);
<> procedure
BitBtn7Click(Sender: TObject);
<> procedure
BitBtn2Click(Sender: TObject);
<> procedure
BitBtn8Click(Sender: TObject);
<> procedure
BitBtn9Click(Sender: TObject);
<> procedure
BitBtn10Click(Sender: TObject);
<> procedure
BitBtn11Click(Sender: TObject);
<> procedure
FormCreate(Sender: TObject);
<> procedure
BitBtn12Click(Sender: TObject);
<> procedure
BitBtn13Click(Sender: TObject);

_
cked;
end;
end.
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
17
{*************************************
\ ********
<u> </u>
{ TeeChart Delphi Component Library }
{ Main Form Demo
{ Main Form Demo
(Conveight (a) 1006 2001 by David
{ Copyright (c) 1996-2001 by David
Berneda }
{ All rights reserved } {***********************************
******
<u> </u>
unit features;
6
interface
uses
SysUtils, WinTypes, WinProcs,
Messages, Classes, Graphics,
Controls,
Forms, Dialogs, Buttons, StdCtrls,
ExtCtrls;
type
TFeaturesForm = class(TForm)
BitBtn5: TBitBtn;
BitBtn6: TBitBtn;
BitBtn7: TBitBtn;
BitBtn3: TBitBtn;
BitBtn9: TBitBtn;
BitBtn11: TBitBtn;
BitBtn10: TBitBtn;
BitBtn2: TBitBtn;
BitBtn8: TBitBtn;
BitBtn12: TBitBtn;
BitBtn13: TBitBtn;
BitBtn14: TBitBtn;
BitBtn4: TBitBtn;
BitBtn15: TBitBtn;
BitBtn16: TBitBtn;
BitBtn17: TBitBtn;
BitBtn18: TBitBtn;
BitBtn19: TBitBtn;
BitBtn20: TBitBtn;
DIDMES, IDMDM,

1.	Z. Name and James
end;	<> procedure
End;	BitBtn14Click(Sender: TObject);
	<> procedure
<> procedure	BitBtn4Click(Sender: TObject);
TFeaturesForm.BitBtn3Click(Sende	<> procedure
r: TObject);	BitBtn15Click(Sender: TObject);
begin	<> procedure
ShowForm(TMiniForm);	BitBtn16Click(Sender: TObject);
end;	<> procedure
	BitBtn17Click(Sender: TObject);
<> procedure	<> procedure
TFeaturesForm.BitBtn5Click(Sende	BitBtn18Click(Sender: TObject);
r: TObject);	<> procedure
begin	BitBtn19Click(Sender: TObject);
ShowForm(TLegendForm);	<> procedure
end;	BitBtn20Click(Sender: TObject);
)	<> procedure
<> procedure	BitBtn21Click(Sender: TObject);
TFeaturesForm.BitBtn6Click(Sende	private
r: TObject);	{ Private declarations }
begin	public
	{ Public declarations }
ShowForm(TBitmapForm);	<> procedure
end;	ShowForm(AFormClass:TFormClas
	*
<> procedure	s);
TFeaturesForm.BitBtn7Click(Sende	end;
r: TObject);	• 1 44
begin	implementation
ShowForm(TDrawForm);	
end;	{\$R *.dfm}
	Uses
<> procedure	UAxisLab,UBitmap,UDraw,ULegen
TFeaturesForm.BitBtn2Click(Sende	d,UMain,UOverBar,MulAxis,
r: TObject);	
begin	UPages, UPrint, UScroll, UShapes, UY
ShowForm(TScrollForm);	Legend, Last Valu, UColor,
end;	
	UMetafil,UAniZoom,UScrollB,UCro
<> procedure	ssH,UKeyboa,LogLab;
TFeaturesForm.BitBtn8Click(Sende	
r: TObject);	<> procedure
begin	TFeaturesForm.ShowForm(AForm
ShowForm(TShapesForm);	Class:TFormClass);
end;	Begin
· · · · · · · · · · · · · · · · · · ·	With AFormClass.Create(Self) do
<> procedure	try
TFeaturesForm.BitBtn9Click(Sende	ShowModal;
r: TObject);	finally
begin	Free;
DEVIII	11009

TFeaturesForm.BitBtn14Click(Send	ShowForm(TOverBarForm);
er: TObject);	end;
begin	chu,
ShowForm(TDigitalForm);	<> procedure
end;	TFeaturesForm.BitBtn10Click(Send
chu,	er: TObject);
<> procedure	begin
TFeaturesForm.BitBtn4Click(Sende	ShowForm(TPrintForm);
r: TObject);	end;
begin	ciiu,
begin	< > nwaaaduwa
ShowForm(TFormAnimatedZoom);	<pre>&lt;&gt; procedure TFeaturesForm.BitBtn11Click(Send</pre>
end;	
enu;	er: TObject);
	begin Show Four (TDa gas Fours)
<> procedure TEastures Form PitPtn15Click(Sand	ShowForm(TPagesForm);
TFeaturesForm.BitBtn15Click(Send	end;
er: TObject);	Z Sware James
Show Form (TM sto Flo Form)	<> procedure
ShowForm(TMetafileForm);	TFeaturesForm.FormCreate(Sender
end;	: TObject);
	begin
<> procedure	if Screen.Width<800 then
TFeaturesForm.BitBtn16Click(Send	ShowMessage('Warning:
er: TObject);	'+#13+#10+
begin	'This Demo is best viewed
ShowForm(TCustomAxisForm);	with a Screen'+#13+#10+
end;	'resolution of 800x600 or
	greater,'+#13+#10+
<> procedure	'and a Color Depth of 256
TFeaturesForm.BitBtn17Click(Send	or greater.'+#13+#10+
er: TObject);	'16K Colors is also better
begin	than 256 Colors.');
ShowForm(TColoredForm);	end;
end;	
	<> procedure
<> procedure	TFeaturesForm.BitBtn12Click(Send
TFeaturesForm.BitBtn18Click(Send	er: TObject);
er: TObject);	begin
begin	ShowForm(TAxisLabelsForm);
ShowForm(TScrollBarForm);	end;
end;	
	<> procedure
<> procedure	TFeaturesForm.BitBtn13Click(Send
TFeaturesForm.BitBtn19Click(Send	er: TObject);
er: TObject);	begin
begin	ShowForm(TLegendXYForm);
ShowForm(TCrossHairForm);	end;
end;	
	<> procedure

LineSeries3: TLineSeries;
LineSeries4: TLineSeries;
LineSeries5: TLineSeries;
PointSeries1: TPointSeries;
Label1: TLabel;
SpinEdit1: TSpinEdit;
BitBtn2: TBitBtn;
BitBtn1: TBitBtn;
BitBtn3: TBitBtn;
Memo1: TMemo;
<> procedure
FormCreate(Sender: TObject);
<> procedure
CheckBox1Click(Sender: TObject);
<> procedure
CheckBox2Click(Sender: TObject); <> procedure
_
SpinEdit1Change(Sender: TObject);
<> procedure P:tPtn1Click(Sondow TObicot)
BitBtn1Click(Sender: TObject);
<> procedure  Pi4Ptn2Click(Sandary TObicet)
BitBtn2Click(Sender: TObject);
private (Private declarations)
{ Private declarations }
public
{ Public declarations }
end;
:
implementation
(CD + 16)
{\$R *.dfm}
<> procedure
TFormAnimatedZoom.FormCreate(
Sender: TObject);
<> procedure
RandomValues(ASeries:TChartSeri
es);
Var t,tmp:Longint;
Begin
With ASeries do
Begin
tmp:=0;
Clear;
for t:=1 to 50 do
Begin
tmp:=tmp+Random(10000)-
5000;

<> procedure
TFeaturesForm.BitBtn20Click(Send
er: TObject);
begin
ShowForm(TKeyboardForm);
end;
<> procedure
TFeaturesForm.BitBtn21Click(Send
er: TObject);
begin
ShowForm(TLogLabelsForm);
end;
enu;
end.
\\\\\\\
18
18
{***********
*******
{ TeeChart. TChart Component
1 eechart. Tenart component
}
{ Copyright (c) 1995-2001 by David
Berneda }
{ All Rights Reserved }
{***********
<u> </u>
unit uanizoom;
interface
uses
SysUtils, WinTypes, WinProcs,
Messages, Classes, Graphics,
Controls,
Forms, Dialogs, Teengine, Series,
ExtCtrls, Chart, StdCtrls, Spin,
Buttons, TeeProcs;
type
TFormAnimatedZoom =
class(TForm)
, ,
Chart1: TChart;
Panel1: TPanel;
CheckBox1: TCheckBox;
CheckBox2: TCheckBox;
·
LineSeries1: TLineSeries;
LineSeries2: TLineSeries;

(Sender: TObject);	Add( tmp, ", clTeeColor );
begin	end;
BitBtn1.Enabled:=False;	end;
BitBtn2.Enabled:=False;	End;
Chart1.ZoomPercent(115);	
BitBtn1.Enabled:=True;	begin
BitBtn2.Enabled:=True;	
end;	SpinEdit1.Value:=Chart1.Animated
	ZoomSteps;
<> procedure	RandomValues(LineSeries1);
TFormAnimatedZoom.BitBtn2Click	RandomValues(LineSeries2);
(Sender: TObject);	RandomValues(LineSeries3);
begin	RandomValues(LineSeries4);
BitBtn1.Enabled:=False;	RandomValues(LineSeries5);
BitBtn2.Enabled:=False;	RandomValues(PointSeries1);
Chart1.ZoomPercent(85);	Chart1.AnimatedZoom:=False;
BitBtn1.Enabled:=True;	Chart1.ZoomPercent(5);
BitBtn2.Enabled:=True;	Chart1.AnimatedZoom:=True;
end;	end;
- Cita,	enu,
end.	<> procedure
	TFormAnimatedZoom.CheckBox1C
111111111111111111111111111111111111111	lick(Sender: TObject);
	begin
19	
	Chart1.AnimatedZoom:=CheckBox1
<b>{*******</b> *****************	.checked;
********	end;
{ TeeChart Delphi Component	,
Library }	<> procedure
{ Gantt Series Type Demo	TFormAnimatedZoom.CheckBox2C
}	lick(Sender: TObject);
{ Copyright (c) 1995-2001 by David	begin
Berneda }	
{ All rights reserved }	Chart1.View3D:=CheckBox2.Check
{*************************************	ed;
********	end;
unit gantt;	
	<> procedure
interface	TFormAnimatedZoom.SpinEdit1Ch
	ange(Sender: TObject);
uses	begin
SysUtils, WinTypes, WinProcs,	
Messages, Classes, Graphics,	Chart1.AnimatedZoomSteps:=SpinE
Controls,	dit1.Value;
Forms, Dialogs, Chart, Series,	end;
ExtCtrls, StdCtrls, GanttCh,	
Teengine, Buttons, TeeProcs;	<> procedure

Use the AddGantt or	type
AddGanttColor methods.	TGanttForm = class(TForm)
	Chart1: TChart;
( This demo do not uses this methods	Panel1: TPanel;
)	Label1: TLabel;
	Shape1: TShape;
Example:	BitBtn3: TBitBtn;
	GanttSeries1: TGanttSeries;
GanttSeries1.AddGantt(	Memo1: TMemo;
EncodeDate( 1997, 1, 1 ),	Label2: TLabel;
EncodeDate(1997, 1,	<> procedure
31),	FormCreate(Sender: TObject);
0,	<> procedure
'Programming');	Chart1MouseMove(Sender:
	TObject; Shift: TShiftState; X,
Or	Y: Integer);
	<> procedure
GanttSeries1.AddGanttColor(	Chart1GetNextAxisLabel(Sender:
EncodeDate( 1997, 1, 1 ),	TChartAxis;
EncodeDate(	LabelIndex: Longint; var
1997, 1, 31),	LabelValue: Double; var Stop:
0,	Boolean);
'Programming',	<> procedure
clGreen );	GanttSeries1Click(Sender:
	TChartSeries; ValueIndex: Integer;
	<b>Button: TMouseButton; Shift:</b>
Where "0" is the desired vertical	TShiftState; X, Y: Integer);
position for this bar.	private
Choose the vertical position you	{ Private declarations }
prefer.	public
	{ Public declarations }
To connect gantt bars:	end;
1) Store the "AddGantt" or	implementation
"AddGanttColor" function return	
longint:	{\$R *.dfm}
Var tmp1, tmp2 : Longint;	Uses UDemUtil;
tmp1:=GanttSeries1.AddGantt(	(*
EncodeDate( 1997, 1, 1 ),	
EncodeDate(	*********
1997, 1, 31 ),	**
0,	HOW TO ADD GANTT BARS
'Programming'	MANUALLY?
);	
	***********
tmp2:=GanttSeries1.AddGantt(	**
EncodeDate( 1997, 4, 1 ),	

eries1.XValues[tmpTask]);	EncodeDate(
eriestat varues (emp ruskj);	1997, 4, 30),
Label2.Caption:=DateToStr(GanttS	0,
eries1.EndValues[tmpTask]);	'Testing');
	8 //
Shape1.Brush.Color:=GanttSeries1.	2) Then use the NextTask
ValueColor[tmpTask];	property:
end	
else	GanttSeries1.NextTask[ tmp1 ] :=
begin { clear label and shape	tmp2;
samples }	
Label1.Caption:='';	This will draw a line from
Label2.Caption:=";	'Programming' gantt bar to
	'Testing' bar.
Shape1.Brush.Color:=Panel1.Color;	
end;	The "ConnectingLinePen"
end;	property is the pen used to draw
	lines.
<> procedure	
TGanttForm.Chart1GetNextAxisLa	*)
bel(Sender: TChartAxis;	
LabelIndex: Longint; var	<> procedure
LabelValue: Double; var Stop:	TGanttForm.FormCreate(Sender:
Boolean);	TObject);
var year,month,day:Word;	begin
this will set the Bottom Axis	GanttSeries1.FillSampleValues(Gant
(Dates) Labels to the First	tSeries1.NumSampleValues); { <
day of each month. }	Some random points }
if Sender=Chart1.BottomAxis then	{ change cursor when mouse moves
begin	over Gantt bars}
{ LabelValue has the "candidate"	GanttSeries1.Cursor:=crTeeHand;
value where the Axis label will be	end;
painted. }	
,	<> procedure
DecodeDate(LabelValue,year,month,	TGanttForm.Chart1MouseMove(Se
day);	nder: TObject; Shift: TShiftState;
{ we force that value to be the first	X, Y: Integer);
day in month }	var tmpTask:Longint;
Day:=1;	begin
Month:=Month+1;	
if Month>12 then	tmpTask:=GanttSeries1.Clicked(x,y)
Begin	; { < if mouse is over a gantt bar
Month:=1;	}
Year:=Year+1;	if tmpTask<>-1 then
end;	begin { set a sample label and a
{ Then we set the preferred Label	sample shape color }
value }	
	Label1.Caption:=DateToStr(GanttS

Massages Classes Cyanhias	LabalValuar-Engada Data(waan mant
Messages, Classes, Graphics,	LabelValue:=EncodeDate(year,mont
Controls,	h,day);
Forms, Dialogs, Teengine, Series,	{ we want more labels !! }
ExtCtrls, Chart, StdCtrls,	Stop:=False;
Buttons, TeeProcs, TeeFunci;	end;
	end;
type	
THighLowForm = class(TForm)	<> procedure
Chart1: TChart;	TGanttForm.GanttSeries1Click(Sen
BarSeries1: TBarSeries;	der: TChartSeries;
Panel1: TPanel;	ValueIndex: Integer; Button:
BitBtn1: TBitBtn;	TMouseButton; Shift: TShiftState;
BitBtn2: TBitBtn;	<b>X</b> ,
CheckBox1: TCheckBox;	Y: Integer);
Timer1: TTimer;	begin
CB3D: TCheckBox;	if Button=mbLeft then
AverageSeries: TLineSeries;	With Sender do
HighSeries: TLineSeries;	With Sender do
LowSeries: TLineSeries;	ValueColor[ValueIndex]:=EditColor
,	(Self, ValueColor[ValueIndex])
CheckBox2: TCheckBox;	else
CheckBox3: TCheckBox;	
Memo1: TMemo;	Chart1.CancelMouse:=False;
<> procedure	{ setting CancelMouse to True
FormCreate(Sender: TObject);	notifies TeeChart engine to STOP
<> procedure	handling the mouse click and
Chart1AfterDraw(Sender:	NOT start zoom or scroll. }
TObject);	end;
<> procedure	_
BitBtn1Click(Sender: TObject);	end.
<> procedure	
CheckBox1Click(Sender: TObject);	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
<> procedure	20
Timer1Timer(Sender: TObject);	{************************
<> procedure	*******
CB3DClick(Sender: TObject);	{ TeeChart Delphi Component
<> procedure	Library }
CheckBox2Click(Sender: TObject);	{ High-Mean-Low Form Demo
<> procedure	}
CheckBox3Click(Sender: TObject);	{ Copyright (c) 1996-2001 by David
private	Berneda }
{ Private declarations }	{ All rights reserved }
public	{**********
{ Public declarations }	\ ************
Bar2:TBarSeries;	unit uhighlo;
· · · · · · · · · · · · · · · · · · ·	umi umgmo,
end;	interface
	mulaut
implementation	NGOG
(OD + 16 )	USES  Carallela Win Tamas Win Due es
{\$R *.dfm}	SysUtils, WinTypes, WinProcs,

;	
	<> procedure
	THighLowForm.FormCreate(Sende
Font.Color:=HighSeries.SeriesColor	r: TObject);
;	begin
TextOut( ChartXCenter,	TeeEraseBack:=False; { try win95
HighSeries.CalcYPos(0),	+ plus! + drag window + resizing!}
HighSeries.Name+'	
'+HighSeries.ValueMarkText[0]);	Bar2:=nil;
	,
	Chart1.View3D:=CB3D.Checked;
Font.Color:=LowSeries.SeriesColor;	Chart1.Chart3DPercent:=35;
TextOut( ChartXCenter,	
LowSeries.CalcYPos(0),	BarSeries1.Fillsamplevalues(6);
LowSeries.Calc 11 0s(0),  LowSeries.Name+'	Dai Sci iest.i insampievalues(0),
	DanGarias 1 Dank 2 D. — Chart 1 Jac
'+LowSeries.ValueMarkText[0]);	BarSeries1.Dark3D:=Chart1.IsScree
end;	nHighColor;
end;	
	Chart1.RightAxis.Minimum:=0;
<> procedure	
THighLowForm.BitBtn1Click(Sende	Chart1.RightAxis.AutomaticMaxim
r: TObject);	um:=True;
begin	BarSeries1.RefreshSeries;
Chart1.PrintLandscape;	end;
end;	
	<> procedure
<> procedure	THighLowForm.Chart1AfterDraw(
THighLowForm.CheckBox1Click(Se	Sender: TObject);
nder: TObject);	begin
begin	if not CheckBox2.Checked then {
begin	only with one bar, much pretty }
Timer1.Enabled:=CheckBox1.Check	With Chart1, Canvas do
ed;	
	begin
end;	Brush.Style:=bsClear;
2 Samuel 1	
<> procedure	
THighLowForm.Timer1Timer(Send	Font.PixelsPerInch:=Screen.PixelsPe
er: TObject);	rInch;
var tmp:Longint;	Font.Size:=12;
begin	Font.Style:=[fsBold,fsItalic];
Timer1.Enabled:=False;	
tmp:=Random(BarSeries1.Count);	
	Font.Color:=AverageSeries.SeriesCo
BarSeries1.YValue[tmp]:=BarSeries	lor;
1.YValue[tmp]+Random(50)-25;	TextOut( ChartXCenter,
if random(100)<8 then	z tato at Cam a zonite,
	AverageSeries.CalcYPos(0),
BarSeries1.BarStyle:=TBarStyle(Ra	AverageSeries.Name+'
ndom(1+Ord(High(TBarStyle))));	S
nuom(1+Oru(mgn(1 DarStyle))));	'+AverageSeries.ValueMarkText[0])

LowSeries.DataSources.Add(Bar2);	{ Randomly change from 3D to 2D
	}
Bar2.AddLinkedSeries(LowSeries);	if random(100)<2 then
end	CB3D.Checked:=not
else	CB3D.Checked;
begin	Timer1.Enabled:=True;
{ remove the second bar series we	end;
created before }	Cita
Bar2.Free;	<> procedure
Bar2:=ric;	THighLowForm.CB3DClick(Sender
end;	: TObject);
,	begin
{ and finally refresh the statistical	Chart1.View3d:=CB3D.Checked;
series to view results }	end;
BarSeries1.RefreshSeries;	enu,
end;	Z Nano coduno
Z. Samuellani	<pre>&lt;&gt; procedure Thigh I ave Form Check Pov2Click (See</pre>
<> procedure	THighLowForm.CheckBox2Click(Se
THighLowForm.CheckBox3Click(Se	nder: TObject);
nder: TObject);	begin
begin	if CheckBox2.Checked then
	begin
BarSeries1.Active:=CheckBox3.Che	{ create a new TBarSeries, and add
cked;	data }
if Assigned(Bar2) then	Bar2:=TBarSeries.Create(Self);
	Bar2.ParentChart:=Chart1;
Bar2.Active:=CheckBox3.Checked;	
end;	Bar2.BarStyle:=bsInvPyramid;
end.	{ fill the second series with same
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	number of random values than
21	the first bar series }
{************************	Bar2.FillSampleValues(
*******	BarSeries1.Count);
{ TeeChart Delphi Component	
Library }	{ then add this new Series as
{ Keyboard Scrolling Demo	datasource for statistics }
}	
{ Copyright (c) 1996-2001 by David	AverageSeries.DataSources.Add(Bar
Berneda }	2);
{ All rights reserved }	
<b>{*****************</b>	Bar2.AddLinkedSeries(AverageSeri
******	es);
unit ukeyboa;	
interface	HighSeries.DataSources.Add(Bar2);
uses	Bar2.AddLinkedSeries(HighSeries);
SysUtils, WinTypes, WinProcs,	

# <u>AC.&P</u>

XRange,YRange:Double;	Messages, Classes, Graphics,
begin	Controls,
{ initialize some temporary	Forms, Dialogs, Teengine, Series,
variables }	ExtCtrls, Chart, StdCtrls, Buttons,
XDelta:=0;	TeeProcs;
YDelta:=0;	
With LineSeries1.GetHorizAxis do	type
XRange:=Maximum-Minimum;	TKeyboardForm = class(TForm)
With LineSeries1.GetVertAxis do	Chart1: TChart;
YRange:=Maximum-Minimum;	LineSeries1: TLineSeries;
{ handle keyboard !!! }	Panel1: TPanel;
if ssShift in Shift then	BitBtn1: TBitBtn;
begin	InvertScroll: TCheckBox;
Case key of	CheckLimits: TCheckBox;
VK LEFT,VK UP :	Memo1: TMemo;
Chart1.ZoomPercent(110);	<> procedure
VK RIGHT, VK DOWN:	FormCreate(Sender: TObject);
Chart1.ZoomPercent(90);	<> procedure
end;	FormKeyDown(Sender: TObject;
exit;	var Key: Word;
end	Shift: TShiftState);
else	<> procedure
Case key of	CheckLimitsClick(Sender:
VK LEFT : XDelta:=-	TObject);
XRange/100;	<> procedure
VK RIGHT : XDelta:=	InvertScrollClick(Sender: TObject);
XRange/100;	private
VK_UP : YDelta:= YRange/100;	{ Private declarations }
VK DOWN : YDelta:=-	public
YRange/100;	{ Public declarations }
vk_Next : YDelta:=-YRange/10;	end;
vk_Prior : YDelta:= YRange/10;	
VK_SPACE : Begin	implementation
Chart1.UndoZoom; Exit; End; { <	
reset scrolling }	{\$R *.dfm}
end;	
{ just to make this example a little	<> procedure
better }	TKeyboardForm.FormCreate(Sende
if not InvertScroll.Checked then	r: TObject);
begin	begin
XDelta:=-XDelta;	LineSeries1.FillSampleValues(500);
YDelta:=-YDelta;	AnimatedZoomFactor:=4;
end;	end;
{ apply scrolling !!! }	
With Chart1 do	<> procedure
Begin	TKeyboardForm.FormKeyDown(Se
	nder: TObject; var Key: Word;
LeftAxis.Scroll(YDelta,CheckLimits.	Shift: TShiftState);
Checked);	Var XDelta, YDelta,

Paintbox component.	
}	RightAxis.Scroll(YDelta,CheckLimit
uses	s.Checked);
SysUtils, WinTypes, WinProcs,	
Messages, Classes, Graphics,	BottomAxis.Scroll(XDelta,CheckLi
Controls,	mits.Checked);
Forms, Dialogs, ExtCtrls, Chart,	
Series, StdCtrls, Teengine, Buttons,	TopAxis.Scroll(XDelta,CheckLimits.
TeeProcs;	Checked);
,	SetFocus;
type	End;
TLegendForm = class(TForm)	end;
Chart1: TChart;	
LineSeries1: TLineSeries;	<> procedure
LineSeries2: TLineSeries;	TKeyboardForm.CheckLimitsClick(
PaintBox1: TPaintBox;	Sender: TObject);
Panel1: TPanel;	begin
BitBtn3: TBitBtn;	ShowMessage('Please zoom before
Label1: TLabel;	scrolling.');
Memo1: TMemo;	Chart1.SetFocus;
<> procedure	end;
LineSeries2AfterDrawValues(Sende	
r: TObject);	<> procedure
<> procedure	TKeyboardForm.InvertScrollClick(
FormCreate(Sender: TObject);	Sender: TObject);
private	begin
{ Private declarations }	Chart1.SetFocus;
public	end;
{ Public declarations }	
end;	end.
var	
LegendForm: TLegendForm;	\\\\
	22
implementation	{*************************************
	******
{\$R *.dfm}	{ TeeChart. Legend Example
	}
<> procedure	{ Copyright (c) 1995-2001 by David
TLegendForm.LineSeries2AfterDra	Berneda }
wValues(Sender: TObject);	{ All Rights Reserved }
var t:Longint;	{****************************
begin	<u></u>
With Paintbox1.Canvas do { we'll	unit ulegend;
draw over PaintBox1 }	
Begin	interface
for t:=0 to Chart1.SeriesCount-1	{ This form shows how to create a
do { for each Series in Chart }	new Chart.Legend.
Begin	We'll draw the Series Titles onto a

Messages, Classes, Graphics,	
Controls,	Font.Color:=Chart1[t].SeriesColor;
Forms, Dialogs, ExtCtrls, Chart,	{ set font color }
	{ draw the customized Series
Series, StdCtrls, Teengine, Buttons,	*
TeeProcs;	Title }
	TextOut(40,20+16*t,'This is a
type	long Series title:
TLegendXYForm = class(TForm)	'+Chart1.SeriesTitleLegend(t));
Chart1: TChart;	end;
PieSeries1: TPieSeries;	End;
Panel1: TPanel;	end;
RadioGroup1: TRadioGroup;	
BitBtn3: TBitBtn;	<> procedure
Memo1: TMemo;	TLegendForm.FormCreate(Sender:
<> procedure	TObject);
FormCreate(Sender: TObject);	begin
<> procedure	LineSeries1.FillSampleValues(50);
Chart1GetLegendRect(Sender:	{ random values }
TCustomChart; var Rect: TRect);	LineSeries2.FillSampleValues(50);
<> procedure	end;
Chart1GetLegendPos(Sender:	
TCustomChart; Index: Longint;	end.
var X, Y, XColor: Longint);	
<> procedure	
RadioGroup1Click(Sender:	24
TObject);	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
private	\ *********
{ Private declarations }	{ TeeChart Delphi Component
public	Library }
{ Public declarations }	{ Custom Legend Size and Position
,	Demo }
DefaultLegend:Boolean;	{ Copyright (c) 1995-2001 by David
end;	Berneda }
Van	{ All rights reserved }
L agand VVE away	{ All rights reserved } { **********************************
LegendXYForm:	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
TLegendXYForm;	<u> </u>
. 1	unit uylegend;
implementation	interfere
(OD 1 14 )	interface
{\$R *.dfm}	{ This form shows a customized
	Legend.
<> procedure	The Chart.OnGetLegendRect and
TLegendXYForm.FormCreate(Send	Chart.OnGetLegendPos events are
er: TObject);	used to
begin	change the default legend size and
DefaultLegend:=False; { <	the default legend text positions.
only used in this example }	}
PieSeries1.FillSampleValues(10); {	uses
< some random pie sectors }	SysUtils, WinTypes, WinProcs,
· some random pie sectors ;	System system system is a second

k(Sender: TObject);	end;
	enu,
begin	
{ Get the RadioGroup selection and	<> procedure
force the chart to repaint }	TLegendXYForm.Chart1GetLegend
	Rect(Sender: TCustomChart;
DefaultLegend:=RadioGroup1.ItemI	var Rect: TRect);
ndex=0;	begin
Chart1.Repaint;	if not DefaultLegend then { < if
end;	we want to customize legend }
	Begin
end.	{ This changes the Legend
	Rectangle dimensions }
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	,
25	Rect.Bottom:=Rect.Top+PieSeries1.
{***********	Count*15; { < Calc Legend Height
\ *********	}
{ TeeChart Delphi Component	Rect.Left:=Rect.Left-120;
Library }	{ < Bigger Legend Width }
{ Linked Tables Chart Demo	end;
{ Linked Tables Chart Demo	,
(C : 14() 1007 2001   D : 1	end;
{ Copyright (c) 1995-2001 by David	
Berneda }	<> procedure
{ All rights reserved }	TLegendXYForm.Chart1GetLegend
{*************************************	Pos(Sender: TCustomChart; Index:
******	Longint;
unit linked;	var X, Y, XColor: Longint);
	begin
interface	if not DefaultLegend then
	Begin
uses	{ Calculate the X Y coordinates
SysUtils, WinTypes, WinProcs,	for each Legend Text }
Messages, Classes, Graphics,	
Controls,	x:=Chart1.Legend.RectLegend.Left;
Forms, Dialogs, StdCtrls, Buttons,	x:=x + (Index div
ExtCtrls, Grids, DBGrids, DB,	(PieSeries1.Count div 2))*100;
DBTables, Chart, Series, DbChart,	( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( )
DBCtrls, Teengine, TeeProcs;	
DDCH is, Teengine, Teerroes,	y:=Chart1.Legend.RectLegend.Top;
type	y:=y + (Index mod
TI in had Tables Forms	(PieSeries1.Count div 2))*30;
TLinkedTablesForm =	(1 ICSCITICS1. COURT UIV 2)) 30,
class(TForm)	(f (Index mod 2)=141 V. V. 20
DBChart1: TDBChart;	if (Index mod 2)=1 then X:=X+20;
DataSource1: TDataSource;	x:=x+20;
DBGrid1: TDBGrid;	XColor:=X-15;
Panel1: TPanel;	end;
Panel2: TPanel;	end;
Table1: TTable;	
Table2: TTable;	<> procedure
DBNavigator1: TDBNavigator;	TLegendXYForm.RadioGroup1Clic

Berneda }	DataSource2: TDataSource;
,	DBGrid2: TDBGrid;
{ All rights reserved } {***********************************	,
*******	Table2OrderNo: TFloatField;
<b>S</b>	Table2CustNo: TFloatField;
unit loglab;	Table2SaleDate: TDateTimeField;
{\$P-}	Table2AmountPaid:
	TCurrencyField;
interface	Table1CustNo: TFloatField;
	Table1Company: TStringField;
{ This form shows how custom Axis	Table1City: TStringField;
labels can be specified }	Table1State: TStringField;
{ The Chart.OnGetNextAxisLabel	Table1Country: TStringField;
event is used to supply the axis with	AreaSeries1: TAreaSeries;
custom label positions and values. }	BitBtn1: TBitBtn;
	<> procedure
uses	DataSource1DataChange(Sender:
WinProcs, WinTypes, Messages,	TObject; Field: TField);
SysUtils, Classes, Graphics,	private
Controls, Forms, Dialogs,	{ Private declarations }
ExtCtrls, TeeProcs, TeEngine,	public
Chart, Series, StdCtrls, Buttons;	{ Public declarations }
	end;
type	CHU,
TLogLabelsForm = class(TForm)	implementation
Chart1: TChart;	implementation
Series1: TFastLineSeries;	{\$R *.dfm}
Panel1: TPanel;	¿sk .uiii;
BitBtn2: TBitBtn;	< > nwoooduwo
Memo1: TMemo;	<> procedure TLinkedTablesForm.DataSource1D
<> procedure	
FormCreate(Sender: TObject);	ataChange(Sender: TObject;
private	Field: TField);
-	begin { force dbchart to refresh! }
{ Private declarations }	DDCL 41 CL LD 4 C (A C
public	DBChart1.CheckDataSource(AreaS
{ Public declarations }	eries1);
end;	end;
implementation	end.
(6D \$ 16)	
{\$R *.dfm}	111111111111111111111111111111111111111
	26
<> procedure	26
TLogLabelsForm.FormCreate(Send	{*************************
er: TObject);	*******
begin	{ TeeChart Delphi Component
{ Axis settings }	Library }
	{ Logarithmic Labels Demo
Chart1.BottomAxis.Logarithmic:=T	}
rue;	{ Copyright (c) 1996-2001 by David

Forms, Dialogs, Teengine, Series,	
ExtCtrls, Chart, StdCtrls,	Chart1.BottomAxis.TickOnLabelsO
Buttons, TeeProcs;	nly:=True;
Dations, Teerroes,	Chart1.BottomAxis.SetMinMax(
type	10.0, 1000);
TMetafileForm = class(TForm)	10.0, 1000),
Chart1: TChart;	Chart1.LeftAxis.Logarithmic:=True
Image1: TImage;	·
BarSeries1: TBarSeries;	,
Panel1: TPanel;	Chart1.LeftAxis.TickOnLabelsOnly:
Panel2: TPanel;	=True;
,	Chart1.LeftAxis.SetMinMax( 10.0,
SaveDialog1: TSaveDialog;	1000);
BitBtn2: TBitBtn;	1000);
BitBtn3: TBitBtn;	{ adding XY values to Series1 }
BitBtn4: TBitBtn;	Series1.XValues.DateTime:=False;
<> procedure	Series1.AddXY( 100, 100, ",
FormCreate(Sender: TObject);	clTeeColor);
<> procedure	Series1.AddXY( 500, 200, ",
BitBtn3Click(Sender: TObject);	clTeeColor);
<> procedure	Series1.AddXY( 800, 300, ",
BitBtn2Click(Sender: TObject);	clTeeColor);
private	,
{ Private declarations }	Series1.AddXY( 200, 200, ", clTeeColor );
public	
{ Public declarations }	end;
end;	1
	end.
implementation	
	111111111111111111111111111111111111111
{\$R *.dfm}	27 (************************************
uses Clipbrd;	***********
	<u> </u>
<> procedure	{ TeeChart Delphi Component
TMetafileForm.FormCreate(Sender:	Library }
TObject);	{ Metafile *.WMF Demo
begin	{C
BarSeries1.FillSampleValues(8); {	{ Copyright (c) 1995-2001 by David
< some sample random bars }	Berneda }
end;	{ All rights reserved }
	{***********************
{ This code copies Chart contents	*******
onto Windows Clipboard in Metafile	unit umetafil;
Format }	
<> procedure	interface
TMetafileForm.BitBtn3Click(Sender	
: TObject);	uses
begin	SysUtils, WinTypes, WinProcs,
{ TeeClipWhenMetafiling:=True; {	Messages, Classes, Graphics,
< FORCE CLIPPING WITH	Controls,

Rect( 0,0, round(21{cm}*37.8),	METAFILES }
round(10{cm}*37.8)));	{ CLIPPING WORKS FINE BUT
*)	DO NOT ALLOW MOVEABLE
{ (this equals to 96 * 21 / 2.54, 96 *	OR RESIZEABLE METAFILES }
10/2.54)	
{ now it's loaded HERE ! }	Chart1.CopyToClipboardMetafile(T
	rue); { < Enhanced Metafile =
Image1.Picture.LoadFromFile(Save	True }
Dialog1.FileName);	
Image1.Refresh;	ShowMessage('Chart1 is now at
end;	Windows Clipboard in Metafile
end;	format.'+#13+
	'and will now be pasted
end.	HERE !');
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	{ Now PASTE! }
28	Image1.Picture.Assign(ClipBoard);
{**********	Image1.Refresh;
*******	end;
{ TeeChart. Mini-Charts Example	
]}	{ This button asks a filename and
{ Copyright (c) 1995-2001 by David	saves the Chart }
Berneda }	<> procedure
{ All Rights Reserved }	TMetafileForm.BitBtn2Click(Sender
{*************************************	: TObject);
*******	begin
unit umain;	if SaveDialog1.Execute then { <
	ask for a filename first }
interface	begin
	{ SAVE IT !! }
{ This project show some very small	
mini-charts and some animation }	{ CLIPPING WORKS FINE BUT
{ That mini-charts could be useful as	DO NOT ALLOW MOVEABLE
small data-monitors in your forms,	OR RESIZEABLE METAFILES }
or reports }	{ TO FORCE CLIPPING WITH
uses	METAFILES UNCOMMENT THIS
SysUtils, WinTypes, WinProcs,	LINE: }
Messages, Classes, Graphics,	{ TeeClipWhenMetafiling:=True;
Controls,	}
Forms, Dialogs, Chart, Series,	
ExtCtrls, Teengine, TeeProcs;	
	Chart1.SaveToMetafile(SaveDialog1
type	.FileName);
TMiniForm = class(TForm)	
LineSeries1: TLineSeries;	{ THIS METHOD CAN BE USED
Panel1: TPanel;	T00:}
Panel2: TPanel;	(*
Chart1: TChart;	Chart1.SaveToMetafileRect(
Chart2: TChart;	SaveDialog1.FileName,

AreaSeries1.FillSampleValues(Num	Chart3: TChart;
Points);	Chart4: TChart;
	PieSeries1: TPieSeries;
LineSeries2.FillSampleValues(NumP	AreaSeries1: TAreaSeries;
oints);	LineSeries2: TLineSeries;
BarSeries1.FillSampleValues(6);	BarSeries1: TBarSeries;
	LineSeries3: TLineSeries;
LineSeries3.FillSampleValues(NumP	Timer1: TTimer;
oints);	<> procedure
end;	FormCreate(Sender: TObject);
	<> procedure
<> procedure	FormResize(Sender: TObject);
TMiniForm.FormResize(Sender:	<> procedure
TObject);	Timer1Timer(Sender: TObject);
begin	<> procedure
{ Equally resize the panels to center	LineSeries2AfterDrawValues(Sende
charts }	r: TObject);
Chart2.Height:=ClientHeight div 2;	<> procedure
Chart3.Height:=ClientHeight div 2;	LineSeries3AfterDrawValues(Sende
Panel1.Width:=ClientWidth div 2;	r: TObject);
end;	private
	{ Private declarations }
<> procedure	public
TMiniForm.Timer1Timer(Sender:	{ Public declarations }
TObject);	PosChart1,PosChart4:Longint; {
	used to draw vertical lines over
{ This <> procedure changes	charts }
the Series values every second }	end;
<> procedure	
RefreshMonitorChart(Chart:TChar	implementation
t; Var PosChart:Longint);	
var t:Longint;	{\$R *.dfm}
LastValueWas:Double;	
Begin	Const NumPoints=30;
Inc(PosChart);	,
if PosChart >= NumPoints then	<> procedure
PosChart:=0;	TMiniForm.FormCreate(Sender:
for t:=0 to Chart.SeriesCount-1 do	TObject);
Begin	begin
if PosChart=0 then	PosChart1:=-1; {
Begin	starting position of vertical divider }
With Chart do { reset scales at	PosChart4:=NumPoints div 2;
the end of monitoring. }	,
Begin	{ Generate some random points }
LeftAxis.Automatic:=True;	
	LineSeries1.FillSampleValues(NumP
LeftAxis.SetMinMax(MinYValue(Le	oints);
ftAxis),MaxYValue(LeftAxis));	PieSeries1.FillSampleValues(8);
end;	•
	•

if PosChart1>=0 then	
With Chart1, Canvas do	LastValueW
Begin	lues.Last;
Pen.Color:=clRed;	end
DoVertLine(	else
Series[0].CalcXPos(PosChart1), { x	LastValueW
}	lue[PosChar
ChartRect.Top+1,	
{ initial Y }	{ change
ChartRect.Bottom-1	random one
{ ending Y }	
);	Chart.Series
end;	LastValueW
end;	plesMax)-
	(6) 46
<> procedure	(ChartSamp
TMiniForm.LineSeries3AfterDrawV	end;
alues(Sender: TObject);	end;
begin	
{ this event draws the blue divider	var tmpPos:
in Chart4 }	begin
if PosChart4>=0 then	D.C. LM
With Chart4, Canvas do	RefreshMon
Begin	hart1); {
Pen.Color:=clBlue;	
DoVertLine(	RefreshMon
Series[0].CalcXPos(PosChart4), { x	
}	hart4); {
ChartRect.Top+1,	With PieSe
{ initial Y }	RotationAng
ChartRect.Bottom-1	mod 359; { 1
{ ending Y }	mou 557, { 1
end;	{ change Ba
end;	With BarSe
enu,	Begin
end.	tmpPos:=
enu.	tilpi os.
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	YValue[tmp
29	(80.0+Rando
	end;
{**********	end;
{ ********	
{ TeeChart Delphi Component	<> pro
Library }	TMiniForm.
{ Overlayed Bars Demo	alues(Sender
} Overlayed Dat's Deliiu	begin
{ Copyright (c) 1995-2001 by David	{ this event
Berneda }	Chart1 }
Defineda (	

LastValueWas:=Chart.Series[t].YVa
lues.Last;
end
else
LastValueWas:=Chart.Series[t].YVa
lue[PosChart-1];
( change the value for a new
{ change the value for a new random one }
random one }
Chart.Series[t].YValue[PosChart]:=
LastValueWas+Random(ChartSam
plesMax)-
(ChartSamplesMax div 2);
end;
end;
var tmpPos:Longint;
begin
RefreshMonitorChart(Chart1,PosC
hart1); { refresh chart1 }
DefreshMenitorChart(Chart4 DesC
RefreshMonitorChart(Chart4,PosC hart4); { refresh chart4 }
narta), {refresh charta}
With PieSeries1 do
RotationAngle:=(RotationAngle+1)
mod 359; { rotate pie }
-
{ change Bar Series values }
With BarSeries1 do
Begin
tmpPos:=Random(Count);
YValue[tmpPos]:=YValue[tmpPos]*
(80.0+Random(40))/100.0;
end;
end;
<> procedure
TMiniForm.LineSeries2AfterDrawV
alues(Sender: TObject);
(this event draws the red divider in
{ this event draws the red divider in Chart! }
Chart1 }

#### <u>AC.&P</u>

SpinEdit1Change(Sender: TObject);	{ All rights reserved }
<> procedure	<b>{*****************</b>
SpinEdit2Change(Sender: TObject);	*******
<> procedure	unit uoverbar;
CBPatternsClick(Sender: TObject);	
<> procedure	interface
BarSeries3Click(Sender:	
TChartSeries; ValueIndex: Integer;	{ This form shows 3 bar series in a
Button: TMouseButton; Shift:	overlayed layout.
TShiftState; X, Y: Integer);	Each Bar Series has a different
private	BarWidthPercent.
{ Private declarations }	The order Series are drawn is the
public	most important thing. See below.
{ Public declarations }	}
<> procedure	uses
ChangeOverBars;	SysUtils, WinTypes, WinProcs,
end;	Messages, Classes, Graphics,
	Controls,
var	Forms, Dialogs, Chart, Series,
OverBarForm: TOverBarForm;	ExtCtrls, Teengine, StdCtrls, Spin,
	<b>Buttons, TeeProcs;</b>
implementation	
	type
{\$R *.dfm}	TOverBarForm = class(TForm)
	Chart1: TChart;
<> procedure	BarSeries1: TBarSeries;
TOverBarForm.FormCreate(Sender	BarSeries2: TBarSeries;
: TObject);	BarSeries3: TBarSeries;
var t:Longint;	Panel1: TPanel;
begin	Memo1: TMemo;
{ lets fill the 3 Bar Series with some	BitBtn3: TBitBtn;
random data }	Label1: TLabel;
{ Series3 has the biggest values,	SpinEdit1: TSpinEdit;
while Series1 has the smaller values	Label2: TLabel;
}	SpinEdit2: TSpinEdit;
With BarSeries3 do	CBPatterns: TCheckBox;
begin	<> procedure
Clear;	FormCreate(Sender: TObject);
for t:=1 to 10 do Add(	<> procedure
100+Random(30),", clTeeColor);	BarSeries2Click(Sender:
end;	TChartSeries; ValueIndex: Integer;
With BarSeries2 do	Button: TMouseButton; Shift:
begin	TShiftState; X, Y: Integer);
Clear;	<> procedure
for t:=1 to 10 do Add(	BarSeries1Click(Sender:
50+Random(10), ", clTeeColor);	TChartSeries; ValueIndex: Integer;
end;	Button: TMouseButton; Shift:
With BarSeries1 do	TShiftState; X, Y: Integer);
begin	<> procedure

mo	C)
TOverBarForm.SpinEdit1Change(S	Clear;
ender: TObject);	for t:=1 to 10 do Add(
begin	20+Random(10), ", clTeeColor);
ChangeOverBars;	end;
end;	ChangeOverBars;
	end;
<> procedure	
TOverBarForm.SpinEdit2Change(S	<> procedure
ender: TObject);	TOverBarForm.BarSeries2Click(Se
begin { increase / decrease the 3	nder: TChartSeries; ValueIndex:
Series OffsetPercent property }	Integer;
{ This will make partially overlay }	Button: TMouseButton; Shift:
partially overlay	TShiftState; X, Y: Integer);
BarSeries3.OffsetPercent:=SpinEdit	begin
2.Value;	Showmessage('You clicked the Red
20,7 41410,9	Bar at point #:
BarSeries1.OffsetPercent:=SpinEdit	'+inttostr(Valueindex));
2.Value-	end;
Round(SpinEdit2.Value*55.0/100.0);	cnu,
Kound(SpinEdit2. v alue"55.0/100.0);	2 Same and an
DanSanias? OffsetDancants—SninEdit	<pre>&lt;&gt; procedure TOwn Pour Pour Pour Pour Pour Pour Pour Pour</pre>
BarSeries2.OffsetPercent:=SpinEdit	TOverBarForm.BarSeries1Click(Se
2.Value-	nder: TChartSeries; ValueIndex:
Round(SpinEdit2.Value*35.0/100.0);	Integer;
end;	Button: TMouseButton; Shift:
	TShiftState; X, Y: Integer);
<> procedure	begin
TOverBarForm.CBPatternsClick(Se	Showmessage('You clicked the
nder: TObject);	Green Bar at point #:
begin	'+inttostr(Valueindex));
if CBPatterns.Checked then	end;
begin	
	<> procedure
BarSeries1.BarBrush.Style:=bsFDia	TOverBarForm.ChangeOverBars;
gonal;	begin { compress / expand the
	overlayed space between bars }
BarSeries2.BarBrush.Style:=bsBDia	
gonal;	BarSeries3.BarWidthPercent:=Spin
	Edit1.Value;
BarSeries3.BarBrush.Style:=bsDiag	,
Cross;	BarSeries1.BarWidthPercent:=Spin
Chart1.BackColor:=clWhite;	Edit1.Value-
end	Round(SpinEdit1.Value*55.0/100.0);
else	Zound(SpinDuitter and Ood/1000);
begin	BarSeries2.BarWidthPercent:=Spin
negiii	Edit1.Value-
Dar Carios 1 Dar Drugh Styles-he Calid	Round(SpinEdit1.Value*35.0/100.0);
BarSeries1.BarBrush.Style:=bsSolid	` • · · · · · · · · · · · · · · · · · ·
5	end;
BarSeries2.BarBrush.Style:=bsSolid	<> procedure

returns the current Chart Page.	<u>;</u>
Chart.NumPages: Longint	
Returns the Total Number of Pages.	BarSeries3.BarBrush.Style:=bsSolid
method Chart.NextPage; Goes	;
to next page.	Chart1.BackColor:=clTeeColor;
method Chart.PreviousPage;	end;
Goes to previous page.	end;
}	
uses	<> procedure
SysUtils, WinTypes, WinProcs,	TOverBarForm.BarSeries3Click(Se
Messages, Classes, Graphics,	nder: TChartSeries;
Controls,	ValueIndex: Integer; Button:
Forms, Dialogs, Chart, Series,	TMouseButton; Shift: TShiftState;
ExtCtrls, StdCtrls, Spin, BubbleCh,	Χ,
Buttons,	Y: Integer);
Teengine, TeeProcs;	begin
	Showmessage('You clicked the
type	Blue Bar at point #:
TPagesForm = class(TForm)	'+inttostr(Valueindex));
Chart1: TChart;	end;
LineSeries1: TLineSeries;	
BubbleSeries1: TBubbleSeries;	end.
Panel1: TPanel;	111111111111111111111111111111111111111
SpinEdit1: TSpinEdit;	30
Label2: TLabel;	<b>{****************</b>
Label1: TLabel;	*******
ButtonPrevious: TBitBtn;	{ TeeChart. TChart Component
ButtonNext: TBitBtn;	}
BLastPage: TBitBtn;	{ Copyright (c) 1995-2001 by David
BFirstPage: TBitBtn;	Berneda }
CheckBox1: TCheckBox;	{ All Rights Reserved }
BitBtn3: TBitBtn;	<b>{****************</b>
Memo1: TMemo;	*******
<> procedure	unit upages;
FormCreate(Sender: TObject);	
<> procedure	interface
SpinEdit1Change(Sender: TObject);	{ This form demonstrates how
<> procedure	Charts can be divided in several
Chart1PageChange(Sender:	Pages.
TObject);	The main Chart property is
<> procedure	"MaxPointsPerPage".
ButtonPreviousClick(Sender:	TeeChart automatically calculates
TObject);	the total number of pages based on
<> procedure	that
ButtonNextClick(Sender: TObject);	property.
<> procedure	The other Paging properties and
CheckBox1Click(Sender: TObject);	methods are:
<> procedure	
BFirstPageClick(Sender: TObject);	Chart.Page : Longint Sets and

Chart1.UndoZoom;	<> procedure
{ show the current page number	BLastPageClick(Sender: TObject);
and the total number of pages }	private
{ (like a report) }	{ Private declarations }
	public
Label1.Caption:=IntToStr(Chart1.P	{ Public declarations }
age)+'/'+IntToStr(Chart1.NumPages	end;
);	
{ enable or disable buttons }	var
	PagesForm: TPagesForm;
ButtonPrevious.Enabled:=Chart1.Pa	, , , , , , , , , , , , , , , , , , ,
ge > 1;	implementation
ButtonNext.Enabled:=Chart1.Page	
< Chart1.NumPages;	{\$R *.dfm}
8 /	(pix .uiii)
BLastPage.Enabled:=ButtonNext.En	< > nrocoduro
abled;	<pre>&lt;&gt; procedure TPagesForm FormCroats(Sonder)</pre>
	TPagesForm.FormCreate(Sender:
BFirstPage.Enabled:=ButtonPreviou	TObject);
s.Enabled;	begin
end;	LineSeries1.FillSampleValues(100);
chu,	{ < some random points }
/ > nwoooduwo	
<> procedure TPagas Form Putton Provious Click(S)	BubbleSeries1.FillSampleValues(100
TPagesForm.ButtonPreviousClick(S	);
ender: TObject);	SpinEdit1.Value:=18; { <
begin	max number of points per page }
Chart1.PreviousPage; { < goto	Chart1PageChange(Chart1); { <
previous chart page }	repaint page / number of pages }
end;	end;
<> procedure	<> procedure
TPagesForm.ButtonNextClick(Send	TPagesForm.SpinEdit1Change(Send
er: TObject);	er: TObject);
begin	begin
Chart1.NextPage; { < goto next	{ change the max number of points
chart page }	per page }
end;	{ a value of Zero means "No pages.
	Show all points" }
<> procedure	,
TPagesForm.CheckBox1Click(Sende	Chart1.MaxPointsPerPage:=SpinEdi
r: TObject);	t1.Value;
begin	Chart1PageChange(Chart1); { <
	repaint page / number of pages }
Chart1.ScaleLastPage:=CheckBox1.	end;
Checked; { < only for last page }	ciiu,
end;	< > nrocodure
	<pre>&lt;&gt; procedure TPagesForm Chart1PageChange(So.)</pre>
<> procedure	TPagesForm.Chart1PageChange(Se
TPagesForm.BFirstPageClick(Sende	nder: TObject);
11 agest of m. Dritstrage Click (Selice	begin

	<u>A</u>
BitBtn3: TBitBtn;	
CheckBox4: TCheckBox;	
Shape1: TShape;	
RadioGroup1: TRadioGroup;	-
CheckBox5: TCheckBox;	-
· ·	
CheckBox6: TCheckBox;	
RadioGroup2: TRadioGroup;	
<> procedure	
FormCreate(Sender: TObject);	
<> procedure	
CheckBox1Click(Sender: TObject);	
<> procedure	
Timer1Timer(Sender: TObject);	
<> procedure	
CheckBox2Click(Sender: TObject);	
<> procedure	7
CheckBox3Click(Sender: TObject);	
<> procedure	1
PieSeries1Click(Sender:	
TChartSeries; ValueIndex: Integer;	
Button: TMouseButton; Shift:	_
TShiftState; X, Y: Integer);	
<> procedure	
CheckBox4Click(Sender: TObject);	_
<> procedure	
Chart1MouseMove(Sender:	
TObject; Shift: TShiftState; X,	
Y: Integer);	
<> procedure	
RadioGroup1Click(Sender:	
TObject);	
<> procedure	
CheckBox5Click(Sender: TObject);	
<> procedure	
CheckBox6Click(Sender: TObject);	
private	
{ Private declarations }	
public	$\exists$
{ Public declarations }	$\dashv$
tmpX,tmpY:Longint;	-
end;	$\dashv$
Citu,	$\dashv$
implementation	$\dashv$
Implementation	$\dashv$
{\$R *.dfm}	$\dashv$
Uses UDemUtil;	$\dashv$
Uses UDemuni;	$\dashv$
< > nwocoduwo	$\dashv$
<> procedure	
TPieForm.FormCreate(Sender:	

r: TObject);
begin
Chart1.Page:=1; { go to first page }
end;
<> procedure
TPagesForm.BLastPageClick(Sende
e
r: TObject);
begin
Chart1.Page:=Chart1.NumPages;
{ go to last page }
end;
enu,
end.
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
31
31 {************
C
*******
{ TeeChart Delphi Component
Library }
{ Pie Series Type Demo
{ The Series Type Demo
}
{ Copyright (c) 1995-2001 by David
Berneda }
{ All rights reserved }
{*************************************
unit pie;
interface
uses
SysUtils, WinTypes, WinProcs,
Messages, Classes, Graphics,
Controls,
Forms, Dialogs, Chart, Series,
ExtCtrls, StdCtrls, Teengine,
Buttons, teeprocs;
Dations, teepi oes,
4
type
TPieForm = class(TForm)
Chart1: TChart;
Panel1: TPanel;
CheckBox1: TCheckBox;
,
Timer1: TTimer;
PieSeries1: TPieSeries;
CheckBox2: TCheckBox;
CheckBox3: TCheckBox;

XRadius+tmpX>Chart1.ChartWidt	TObject);
-	
h div 3 then tmpX:=-tmpX; end	begin
	tmpX:=-4;
else if XRadius+tmpX<30 then	tmpY:=-3;
tmpX:=-tmpX;	PieSeries1.FillSampleValues(8); {
	< Some random points }
CustomXRadius:=XRadius+tmpX;	
	Chart1.Gradient.Visible:=Chart1.Is
if tmpY>0 then	ScreenHighColor;
Begin	if Chart1.Gradient.Visible then
if	Chart1.Title.Font.Color:=clWhite
YRadius+tmpY>Chart1.ChartHeigh	else
t div 3 then tmpY:=-tmpY;	Chart1.Title.Font.Color:=clBlue;
end	
else	CheckBox6.Checked:=Chart1.Gradi
if YRadius+tmpY<30 then	ent.Visible;
tmpY:=-tmpY;	end;
	,
CustomYRadius:=YRadius+tmpY;	<> procedure
end;	TPieForm.CheckBox1Click(Sender:
	TObject);
{ Change Marks.Font.Color	begin
randomly }	begin
{ if Random(100)<4 then	Timer1.Enabled:=CheckBox1.Check
With PieSeries1.Marks do	ed; { < animation on/off }
	CheckBox3.Enabled:=not
Begin	Timer1.Enabled;
Repeat	if not Timer1.Enabled then
BackColor:=ColorPalette[1+Rando	Begin No. 1 Co. 1 NO. 11 Co. 1
m(MaxDefaultColors)];	PieSeries1.CustomXRadius:=0;
	PieSeries1.CustomYRadius:=0;
Font.Color:=ColorPalette[1+Rando	end;
m(MaxDefaultColors)];	end;
Until BackColor > Font.Color;	
	<> procedure
Arrow.Color:=ColorPalette[1+Rand	TPieForm.Timer1Timer(Sender:
om(MaxDefaultColors)];	TObject);
end;	begin
	{ increase pie rotation angle to get
{ Change Legend positioning	animation }
Randomly }	PieSeries1.Rotate(5);
if Random(100)<4 then	
	{ change X and Y radius to get
RadioGroup1.ItemIndex:=Random(	animation }
RadioGroup1.Items.Count);	With PieSeries1 do
end;	Begin
<u> </u>	if tmpX>0 then
<> procedure	Begin
TPieForm.CheckBox2Click(Sender:	if
11 101 01 m. Checkboazenen behaelt	11

begin	TObject);
tmp:=PieSeries1.Clicked(x,y);	begin
if tmp=-1 then	begin
Shape1.Visible:=False	Chart1.View3d:=CheckBox2.Checke
•	
else	d; { < 3d on / off }
begin	end;
Shape1.Visible:=True;	
	<> procedure
Shape1.Brush.Color:=PieSeries1.Val	TPieForm.CheckBox3Click(Sender:
ueColor[tmp];	TObject);
end;	begin
end;	
	PieSeries1.Circled:=CheckBox3.Che
<> procedure	<pre>cked; { &lt; circled / elliptic pie }</pre>
TPieForm.RadioGroup1Click(Sende	end;
r: TObject);	- ")
begin	<> procedure
Degin .	TPieForm.PieSeries1Click(Sender:
Chartt Logand Alignments—TLeggs	TChartSeries;
Chart1.Legend.Alignment:=TLegen	ValueIndex: Integer; Button:
dAlignment(RadioGroup1.ItemInde	9 ,
x);	TMouseButton; Shift: TShiftState;
end;	X,
	Y: Integer);
<> procedure	begin
TPieForm.CheckBox5Click(Sender:	{ On Clicked Pie, let the user
TObject);	<pre>change the clicked pie color }</pre>
begin	With PieSeries1 do
PieSeries1.Marks.Visible:=CheckBo	ValueColor[ValueIndex]:=EditColor
x5.Checked;	(Self, ValueColor[ValueIndex]);
end;	end;
chu,	Cita;
< > nrocoduro	<> procedure
> procedure	TPieForm.CheckBox4Click(Sender:
TPieForm.CheckBox6Click(Sender:	`
TObject);	TObject);
begin	begin
	Did the second
Chart1.Gradient.Visible:=CheckBox	PieSeries1.UsePatterns:=CheckBox4
6.Checked;	.Checked;
end;	if PieSeries1.UsePatterns then
	PieSeries1.CircleBackColor:=clWhit
end.	e;
	end;
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	
33	<> procedure
\{\psi \psi \psi \psi \psi \psi \psi \psi	TPieForm.Chart1MouseMove(Sende
\   ***********	r: TObject; Shift: TShiftState; X,
}	Y: Integer);
{ TeeChart Delphi Component	
Library }	var tmp:Longint;

<> procedure
TPrintForm.FormCreate(Sender:
TObject);
begin
LineSeries1.FillSampleValues(30);
1
{ < we need some random values }
end;
<> procedure
TPrintForm.BitBtn1Click(Sender:
TObject);
var h,w:longint;
begin
Screen.Cursor := crHourGlass; { <-
- nice detail }
try
Printer.BeginDoc; { < start
<pre>printer job }</pre>
try
{ now print some text on
printer.canvas }
With Printer.Canvas do
begin
Font.Name:='Arial';
Font.Size:=10; { < set
the font size }
Font.Style:=[];
TextOut(0,0,Edit1.Text); { <
<pre>print some text }</pre>
end;
Chu,
ha-Dwinton Donallai-late ( a 4
h:=Printer.PageHeight; { < get
page height }
w:=Printer.PageWidth; { < get
page width }
{ And now print the chart
component }
Chart1.PrintPartial( Rect( w
div 10, { < left margin }
h div 3, {
< top margin }
w - (w div 10),
{ < right margin }
h - (h div 10) ));
{ < bottom margin }
g )
(maint mana tant
{ print more text }

{ Mixed Text and Chart Print Demo }  { Copyright (c) 1995-2001 by David Berneda }  { All rights reserved }  {***********************************	
Serneda   Solution   Served   Solution   Served   Solution   Served   Solution   Served   S	{ Mixed Text and Chart Print Demo
{All rights reserved {************************************	' = ' = ' '
{*************************************	{ All rights reserved }
unit uprint;  { This example shows how to print both Text and Chart in the SAME PAGE }  interface  uses  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,  Forms, Dialogs, StdCtrls, Buttons, Chart, Series, ExtCtrls, Teengine, TeeProcs;  type  TPrintForm = class(TForm)  Chart1: TChart;  LineSeries1: TLineSeries;  BitBtn1: TBitBtn;  Edit1: TEdit;  BitBtn2: TBitBtn;  Label1: TLabel;  <> procedure  FormCreate(Sender: TObject); > procedure  BitBtn1Click(Sender: TObject);  private  { Private declarations }  public  { Public declarations }  end;  var  PrintForm: TPrintForm;  implementation  {\$R *.dfm}	{***********************
{This example shows how to print both Text and Chart in the SAME PAGE }  interface  uses  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,  Forms, Dialogs, StdCtrls, Buttons, Chart, Series, ExtCtrls, Teengine,  TeeProcs;  type  TPrintForm = class(TForm)  Chart1: TChart;  LineSeries1: TLineSeries;  BitBtn1: TBitBtn;  Edit1: TEdit;  BitBtn2: TBitBtn;  Label1: TLabel;  <> procedure  FormCreate(Sender: TObject);  private  { Private declarations }  public  { Public declarations }  end;  var  PrintForm: TPrintForm;  implementation  {\$R *.dfm}	******
both Text and Chart in the SAME PAGE }  interface  uses SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Buttons, Chart, Series, ExtCtrls, Teengine, TeeProcs;  type TPrintForm = class(TForm) Chart1: TChart; LineSeries1: TLineSeries; BitBtn1: TBitBtn; Edit1: TEdit; BitBtn2: TBitBtn; Label1: TLabel; <> procedure FormCreate(Sender: TObject); > procedure BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm;  implementation  {\$R *.dfm}	unit uprint;
both Text and Chart in the SAME PAGE }  interface  uses SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Buttons, Chart, Series, ExtCtrls, Teengine, TeeProcs;  type TPrintForm = class(TForm) Chart1: TChart; LineSeries1: TLineSeries; BitBtn1: TBitBtn; Edit1: TEdit; BitBtn2: TBitBtn; Label1: TLabel; <> procedure FormCreate(Sender: TObject); > procedure BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm;  implementation  {\$R *.dfm}	This example shows how to print
interface  uses  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Buttons, Chart, Series, ExtCtrls, Teengine, TeeProcs;  type  TPrintForm = class(TForm)  Chart1: TChart; LineSeries1: TLineSeries; BitBtn1: TBitBtn; Edit1: TEdit; BitBtn2: TBitBtn; Label1: TLabel; <> procedure FormCreate(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm;  implementation  {\$R *.dfm}	
uses SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Buttons, Chart, Series, ExtCtrls, Teengine, TeeProcs;  type TPrintForm = class(TForm) Chart1: TChart; LineSeries1: TLineSeries; BitBtn1: TBitBtn; Edit1: TEdit; BitBtn2: TBitBtn; Label1: TLabel; <> procedure FormCreate(Sender: TObject);> procedure BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm; implementation  {\$R *.dfm}	PAGE }
uses SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Buttons, Chart, Series, ExtCtrls, Teengine, TeeProcs;  type TPrintForm = class(TForm) Chart1: TChart; LineSeries1: TLineSeries; BitBtn1: TBitBtn; Edit1: TEdit; BitBtn2: TBitBtn; Label1: TLabel; <> procedure FormCreate(Sender: TObject);> procedure BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm; implementation  {\$R *.dfm}	
SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Buttons, Chart, Series, ExtCtrls, Teengine, TeeProcs;  type TPrintForm = class(TForm) Chart1: TChart; LineSeries1: TLineSeries; BitBtn1: TBitBtn; Edit1: TEdit; BitBtn2: TBitBtn; Label1: TLabel; <> procedure FormCreate(Sender: TObject); <> procedure BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm; implementation  {\$R *.dfm}	interface
SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Buttons, Chart, Series, ExtCtrls, Teengine, TeeProcs;  type TPrintForm = class(TForm) Chart1: TChart; LineSeries1: TLineSeries; BitBtn1: TBitBtn; Edit1: TEdit; BitBtn2: TBitBtn; Label1: TLabel; <> procedure FormCreate(Sender: TObject); <> procedure BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm; implementation  {\$R *.dfm}	uses
Controls, Forms, Dialogs, StdCtrls, Buttons, Chart, Series, ExtCtrls, Teengine, TeeProcs;  type TPrintForm = class(TForm) Chart1: TChart; LineSeries1: TLineSeries; BitBtn1: TBitBtn; Edit1: TEdit; BitBtn2: TBitBtn; Label1: TLabel; <> procedure FormCreate(Sender: TObject); <> procedure BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm;  implementation  {\$R *.dfm}	
Forms, Dialogs, StdCtrls, Buttons, Chart, Series, ExtCtrls, Teengine, TeeProcs;  type TPrintForm = class(TForm) Chart1: TChart; LineSeries1: TLineSeries; BitBtn1: TBitBtn; Edit1: TEdit; BitBtn2: TBitBtn; Label1: TLabel; <> procedure FormCreate(Sender: TObject); <> procedure BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm;  implementation  {\$R *.dfm}	
Chart, Series, ExtCtrls, Teengine, TeeProcs;  type TPrintForm = class(TForm) Chart1: TChart; LineSeries1: TLineSeries; BitBtn1: TBitBtn; Edit1: TEdit; BitBtn2: TBitBtn; Label1: TLabel; <> procedure FormCreate(Sender: TObject); <> procedure BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm;  implementation  {\$R *.dfm}	· · · · · · · · · · · · · · · · · · ·
type TPrintForm = class(TForm) Chart1: TChart; LineSeries1: TLineSeries; BitBtn1: TBitBtn; Edit1: TEdit; BitBtn2: TBitBtn; Label1: TLabel; <> procedure FormCreate(Sender: TObject); <> procedure BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm;  implementation  {\$R *.dfm}	
type TPrintForm = class(TForm) Chart1: TChart; LineSeries1: TLineSeries; BitBtn1: TBitBtn; Edit1: TEdit; BitBtn2: TBitBtn; Label1: TLabel; <> procedure FormCreate(Sender: TObject); <> procedure BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm;  implementation  {\$R *.dfm}	
TPrintForm = class(TForm) Chart1: TChart; LineSeries1: TLineSeries; BitBtn1: TBitBtn; Edit1: TEdit; BitBtn2: TBitBtn; Label1: TLabel; <> procedure FormCreate(Sender: TObject); <> procedure BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm;  implementation  {\$R *.dfm}	10011000,
Chart1: TChart; LineSeries1: TLineSeries; BitBtn1: TBitBtn; Edit1: TEdit; BitBtn2: TBitBtn; Label1: TLabel; <> procedure FormCreate(Sender: TObject); <> procedure BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm;  implementation  {\$R *.dfm}	
LineSeries1: TLineSeries; BitBtn1: TBitBtn; Edit1: TEdit; BitBtn2: TBitBtn; Label1: TLabel; <> procedure FormCreate(Sender: TObject); <> procedure BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm;  implementation  {\$R *.dfm}	` /
BitBtn1: TBitBtn; Edit1: TEdit; BitBtn2: TBitBtn; Label1: TLabel; <> procedure FormCreate(Sender: TObject); <> procedure BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm;  implementation  {\$R *.dfm}	· · · · · · · · · · · · · · · · · · ·
Edit1: TEdit; BitBtn2: TBitBtn; Label1: TLabel; <> procedure FormCreate(Sender: TObject); <> procedure BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm;  implementation  {\$R *.dfm}	
Label1: TLabel;  <> procedure FormCreate(Sender: TObject);  <> procedure BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm;  implementation  {\$R *.dfm}	
<> procedure FormCreate(Sender: TObject); <> procedure BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm;  implementation  {\$R *.dfm}	BitBtn2: TBitBtn;
FormCreate(Sender: TObject);  <> procedure  BitBtn1Click(Sender: TObject);  private { Private declarations }  public { Public declarations }  end;  var  PrintForm: TPrintForm;  implementation  {\$R *.dfm}	·
<pre>&lt;&gt; procedure BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm; implementation  {\$R *.dfm}</pre>	
BitBtn1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm;  implementation  {\$R*.dfm}	
private { Private declarations } public { Public declarations } end;  var PrintForm: TPrintForm; implementation  {\$R*.dfm}	
public { Public declarations } end;  var PrintForm: TPrintForm;  implementation  {\$R *.dfm}	
{ Public declarations } end;  var PrintForm: TPrintForm;  implementation  {\$R *.dfm}	
end; var PrintForm: TPrintForm; implementation  {\$R *.dfm}	
var PrintForm: TPrintForm; implementation  {\$R *.dfm}	
PrintForm: TPrintForm; implementation  {\$R *.dfm}	enu;
implementation  {\$R *.dfm}	var
{\$R *.dfm}	PrintForm: TPrintForm;
{\$R *.dfm}	
-	implementation
-	{\$R *.dfm}
· • · · · · · · · · · · · · · · · · · ·	

Series, ExtCtrls, Chart, TeeProcs;
tyna
type To the transfer of the tr
TScrollBarForm = class(TForm)
Chart1: TChart;
LineSeries1: TLineSeries;
ScrollBar2: TScrollBar;
ScrollBar1: TScrollBar;
<> procedure
ScrollBar1Change(Sender:
TObject);
<> procedure
FormCreate(Sender: TObject);
<> procedure
ScrollBar2Change(Sender:
TObject);
<> procedure
Chart1Scroll(Sender: TObject);
<> procedure
Chart1Zoom(Sender: TObject);
<> procedure
Chart1UndoZoom(Sender:
•
TObject);
<pre>&lt;&gt; procedure ChartIP asize(Sandam TObicat);</pre>
Chart1Resize(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
ChangingBars:Boolean;
<> procedure
CalcScrollBarPos;
end;
implementation
{\$R *.dfm}
<> procedure
TScrollBarForm.ScrollBar1Change(
Sender: TObject);
var Difer:Double;
begin
if not ChangingBars then
With Chart1.BottomAxis do
Begin M:
Difer:=Maximum-Minimum;
Maximum:=Chart1.MaxXValue(Ch

With Printer.Canvas do
begin
Font.Name:='Arial';
Font.Size:=12; { < set
the font size }
Font.Style:=[fsItalic];
TextOut(0,60,Edit1.Text+'
again'); { < print some text }
end;
Printer.EndDoc; { < end job
and print !! }
except
on Exception do { just in case an
error happens }
Begin
Printer.Abort;
Printer.EndDoc;
Raise; { < raise up the
exception !!! }
end;
end;
finally
Screen.Cursor:=crDefault; { <
restore cursor }
end;
end;
end.
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
34
{***********
******
{ TeeChart }
{ Scroll Bars Example }
{ Copyright (c) 1995-2001 by David
Berneda }
{ All Rights Reserved }
{**********
*********
******
<b>S</b>
unit uscrollb;
unit uscrollb;
<b>S</b>
unit uscrollb; interface
unit uscrollb; interface uses
unit uscrollb; interface uses SysUtils, WinTypes, WinProcs,
unit uscrollb; interface uses SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics,
unit uscrollb; interface uses SysUtils, WinTypes, WinProcs,

	art1.BottomAxis)-
Round(100.0*(BottomAxis.Minimu	ScrollBar1.Position*Difer/100.0;
m-MinXValue(BottomAxis))/Difer);	Minimum:=Maximum-Difer;
end;	end;
if LeftAxis.Automatic then	end;
ScrollBar2.Enabled:=False	
else	<> procedure
Begin	TScrollBarForm.FormCreate(Sende
ScrollBar2.Enabled:=True;	r: TObject);
Difer:=MaxYValue(LeftAxis)-	begin
MinYValue(LeftAxis);	
if Difer>0 then	LineSeries1.FillSampleValues(1000);
ScrollBar2.Position:=	Chart1.ZoomPercent(115);
12 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	CalcScrollBarPos;
Round(100.0*(LeftAxis.Minimum-	end;
MinYValue(LeftAxis))/Difer);	
end;	<> procedure
end;	TScrollBarForm.ScrollBar2Change(
ChangingBars:=False;	Sender: TObject);
End;	Var Difer:Double;
	begin
<> procedure	if not ChangingBars then
TScrollBarForm.Chart1Scroll(Send	With Chart1.LeftAxis do
er: TObject);	Begin
begin	Difer:=Maximum-Minimum;
CalcScrollBarPos;	Ditti: Maaimam Minimam,
end;	Minimum:=Chart1.MinYValue(Cha
	rt1.LeftAxis)+ScrollBar2.Position*D
<> procedure	ifer/100.0;
TScrollBarForm.Chart1Zoom(Send	Maximum:=Minimum+Difer;
er: TObject);	end;
begin	end;
CalcScrollBarPos;	
end;	<> procedure
onu,	TScrollBarForm.CalcScrollBarPos;
<> procedure	Var Difer:Double;
TScrollBarForm.Chart1UndoZoom(	Begin
Sender: TObject);	ChangingBars:=True;
begin	With Chart1 do
CalcScrollBarPos;	Begin
end;	if BottomAxis.Automatic then
	ScrollBar1.Enabled:=False
<> procedure	else
TScrollBarForm.Chart1Resize(Send	Begin
er: TObject);	ScrollBar1.Enabled:=True;
begin	Difer:=MaxXValue(BottomAxis)-
ScrollBar1.Left:=0;	MinXValue(BottomAxis);
ScrollBar1.Top:=Chart1.Height-	if Difer>0 then
ScrollBar1.Height;	ScrollBar1.Position:=
~ vi viibui i iiivigiiti	SCI UIIDAI 1.1 USILIUII.—

### <u>AC.&P</u>

ScrollBar1.Width:=Chart1.Width;  CBVerticalClick(Sender: TObject); private { Private declarations } public { Public declarations } } public { Public declarations }		
CBVerticalClick(Sender: TObject); private {Private declarations } public {Public declarations }  C——> procedure AddPoint(Const x,y:Double; AColor: TColor);  C——> procedure FillDemoPoints; end;  Implementation  {SR *.dfm}  {This is the event we need to arrange Axis scale as new points are added.}  C——> procedure TScrollForm.LineSeries1AfterAdd(Sender: TChartSeries; ValueIndex: Longint); begin {If VEETICAL SCROLL} if CBVertical.Checked then begin Automatic := False; {	<u> </u>	ScrollBar1.Width:=Chart1.Width;
CBVerticalClick(Sender: TObject); private { Private declarations } public { Public declarations } <	· · · · · · · · · · · · · · · · · · ·	
Private   { Private declarations }   public   { Public declarations }	-	
{ Private declarations }     public     { Public declarations }     { Public declarations }     { Composedure }     AddPoint(Const x,y:Double;     AColor:TColor);     { Composedure }     FillDemoPoints;     end;     { Implementation }     { SerollBar2.Height:=Chart1.Height;     end;     { Composedure }     { Color:TColor);     { Composedure }     { SerollBar2.Height:=Chart1.Height;     end;     { Color:TColor);     { Composedure }     { SerollBar2.Height:=Chart1.Height;     end;     { Cond. }     { SerollBar2.Height:=Chart1.Height;     end;     { Cond. }     { SerollBar2.Height:=Chart1.Height;     end;     { SerollBar2.Height:=Chart1.Height;     end;     { SerollBar2.Height:=Chart1.Height;     end;     { SerollBar2.Height:=Chart1.Height;     end;     { end. }     { SerollBar2.Height:=Chart1.Height;     end;     { SerollBar2.Height:=Chart1.Height;     { SerollBar2.Height:=Chart1.Height;     { SerollBar2.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.Height:=Chart1.He	CBVerticalClick(Sender: TObject);	ScrollBar2.Width;
public { Public declarations } <	private	ScrollBar2.Top:=0;
{ Public declarations }	{ Private declarations }	ScrollBar2.Height:=Chart1.Height;
cnd.	public	end;
AddPoint(Const x,y:Double; AColor:TColor);  <	{ Public declarations }	,
AddPoint(Const x,y:Double; AColor:TColor);  <	<> procedure	end.
AColor:TColor);  <> procedure FillDemoPoints; end;  implementation  {SR *.dfm}  {This is the event we need to arrange Axis scale as new points are added.}  {> procedure TScrollForm.LineSeries1AfterAdd(Sender: TChartSeries; ValueIndex: Longint); begin  With Sender.GetVertAxis do {		
<pre> &lt;&gt; procedure FillDemoPoints; end;  implementation  {SR *.dfm}  {This is the event we need to arrange Axis scale as new points are added.}  &lt;&gt; procedure TScrollForm.LineSeries1AfterAdd(S ender: TChartSeries;  ValueIndex: Longint); begin {If VERTICAL SCROLL} if CBVertical.Checked then begin With Sender.GetVertAxis do {&lt; with the Vertical Axis} Begin Automatic := False; {&lt; wedont want automatic scaling}  {In this example, we will set the Axis Minimum and Maximum values to show One Hour of data ending at last point Time plus 5 minutes }  Minimum := 0; Maximum := Sender.YValues.MaxValue + DateTimeStep[dtTiveMinutes]; Minimum := Maximum DateTimeStep[dtOneHour];  ***********************************</pre>		
FillDemoPoints; end;  (miplementation  {SR *.dfm}  {This is the event we need to arrange Axis scale as new points are added.}  <> procedure TScrollForm.LineSeries1AfterAdd(Sender: TChartScries; ValueIndex: Longint); begin {If VERTICAL SCROLL} if CBVertical.Checked then begin Automatic := False; {< we dont want automatic scaling}  {In this example, we will set the Axis Minimum and Maximum values to show One Hour of data ending at last point Time plus 5 minutes } Minimum := 0; Maximum := Sender.YValues.MaxValue + DateTimeStep[ dtFiveMinutes ]; Minimum := Maximum - DateTimeStep[ dtOneHour ];  ***********************************	· · · · · · · · · · · · · · · · · · ·	
controls	-	\\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\
{TecChart Delphi Component Library } {Auto Axis scaling Demo} {Copyright (c) 1995-2001 by David Berneda } {All rights reserved reserved reserved reserved reserved reserved reserved re	,	1
This is the event we need to arrange Axis scale as new points are added. }	Citat	}
{Auto Axis scaling Demo}  {This is the event we need to arrange Axis scale as new points are added.}  <> procedure TScrollForm.LineSeries1AfterAdd(S ender: TChartSeries;  ValueIndex: Longint); begin  {If VERTICAL SCROLL}  if CBVertical.Checked then begin  With Sender.GetVertAxis do {< with the Vertical Axis}  Begin  Automatic := False; {< with the Vertical Axis (Sender Series)    Automatic := False; {< with axis Minimum and Maximum values to  In this example, we will set the Axis Minimum and Maximum values to  show One Hour of data ending at last point Time plus 5 minutes }  Minimum := 0;  Maximum := Sender.YValues.MaxValue + DateTimeStep[ dtFiveMinutes ];  Minimum := Maximum DateTimeStep[ dtOneHour ];   {Auto Axis scaling Demo }  {Copyright (c) 1995-2001 by David Berneda }  {All rights reserved }  {Automatic selvertAxis do {<	implementation	` -
{ This is the event we need to arrange Axis scale as new points are added. }	Implementation	,
{ This is the event we need to arrange Axis scale as new points are added.}  <> procedure     TScrollForm.LineSeries1AfterAdd(Sender: TChartSeries;     ValueIndex: Longint);     begin     { If VERTICAL SCROLL }     if CBVertical.Checked then     begin     Automatic := False;     dont want automatic scaling }      { In this example, we will set the Axis Minimum and Maximum values to         show One Hour of data ending at last point Time plus 5 minutes     }     Minimum := 0;     Maximum :=     Sender.YValues.MaxValue +     DateTimeStep[ dtFiveMinutes ];     Minimum := Maximum -     DateTimeStep[ dtOneHour ];     { Copyright (c) 1995-2001 by David Berneda }     { All rights reserved }     { Autouscroll;     interface     uses     SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,     Forms, Dialogs, StdCtrls, Chart, Series, ExtCtrls, Teengine, Buttons,     TeeProcs;     Date TimeStep[ dtFiveMinutes ];     Minimum := 0;     Maximum := (and the procedure in the plus of	(CD * dfm)	Auto Axis scaling Demo
{This is the event we need to arrange Axis scale as new points are added.}  <> procedure TScrollForm.LineSeries1AfterAdd(S ender: TChartSeries;  ValueIndex: Longint); begin {If VERTICAL SCROLL} if CBVertical.Checked then begin With Sender.GetVertAxis do { with the Vertical Axis} Begin Automatic := False; { we dont want automatic scaling}  {In this example, we will set the Axis Minimum and Maximum values to     show One Hour of data ending at last point Time plus 5 minutes } Minimum := 0; Maximum := Sender.YValues.MaxValue + DateTimeStep[ dtFiveMinutes ]; Minimum := Maximum - DateTimeStep[ dtOneHour ];  Berneda }  {All rights reserved }  {***********************************	{5K ".uiiii}	(C '14() 1005 2001   D '1
arrange Axis scale as new points are added. }  <> procedure TScrollForm.LineSeries1AfterAdd(S ender: TChartSeries;  ValueIndex: Longint); begin {If VERTICAL SCROLL} if CBVertical.Checked then begin  With Sender.GetVertAxis do {< with the Vertical Axis} Begin  Automatic := False; {< we dont want automatic scaling}  {In this example, we will set the Axis Minimum and Maximum values to		· 10 0 ()
> procedure TScrollForm.LineSeries1AfterAdd(S ender: TChartSeries;  ValueIndex: Longint); begin  {If VERTICAL SCROLL } if CBVertical.Checked then begin  With Sender.GetVertAxis do { with the Vertical Axis }  Begin  Automatic := False; { we dont want automatic scaling }  {In this example, we will set the Axis Minimum and Maximum values to  show One Hour of data ending at last point Time plus 5 minutes }  Minimum := 0;  Maximum := Sender.YValues.MaxValue + DateTimeStep[ dtFiveMinutes ];  Minimum := Maximum DateTimeStep[ dtOneHour ];  **************  unit uscroll;  interface  **************  unit uscroll;  interface  **************  unit uscroll;  interface  **************  **************  unit uscroll;  interface  ***************  *************  ******		,
> procedure TScrollForm.LineSeries1AfterAdd(S ender: TChartSeries;  ValueIndex: Longint); begin  {If VERTICAL SCROLL } if CBVertical.Checked then begin  With Sender.GetVertAxis do { with the Vertical Axis }  Begin  Automatic := False; { we dont want automatic scaling }  {In this example, we will set the Axis Minimum and Maximum values to  show One Hour of data ending at last point Time plus 5 minutes }  Minimum := 0;  Maximum := Sender.YValues.MaxValue + DateTimeStep[ dtFiveMinutes ];  Minimum := Maximum DateTimeStep[ dtOneHour ];  **************  unit uscroll;  interface  **************  unit uscroll;  interface  **************  unit uscroll;  interface  **************  **************  unit uscroll;  interface  ***************  *************  ******	<u> </u>	{ All rights reserved }
TScrollForm.LineSeries1AfterAdd(Sender: TChartSeries;  ValueIndex: Longint); begin  {If VERTICAL SCROLL } if CBVertical.Checked then begin  With Sender.GetVertAxis do {< with the Vertical Axis }  Begin  Automatic := False; {< we dont want automatic scaling }  {In this example, we will set the Axis Minimum and Maximum values to     show One Hour of data ending at last point Time plus 5 minutes }  Minimum := 0;  Maximum := Sender.YValues.MaxValue + DateTimeStep[ dtFiveMinutes ]; Minimum := Maximum DateTimeStep[ dtOneHour ];  winit uscroll; interface  uses SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Chart, Series, ExtCtrls, Teengine, Buttons, TeeProcs;  Chart1: TChart; LineSeries1: TLineSeries; Panel1: TPanel; BitBtn3: TBitBtn; CBVertical: TCheckBox; <> procedure LineSeries1AfterAdd(Sender: TChartSeries; ValueIndex: Longint); <> procedure	,	
ender: TChartSeries;  ValueIndex: Longint); begin  {If VERTICAL SCROLL} if CBVertical.Checked then begin  With Sender.GetVertAxis do {< with the Vertical Axis}  Begin  Automatic := False; {< we dont want automatic scaling }  {In this example, we will set the Axis Minimum and Maximum values to  show One Hour of data ending at last point Time plus 5 minutes }  Minimum := 0;  Maximum := Sender.YValues.MaxValue + DateTimeStep[ dtFiveMinutes ];  Minimum := Maximum DateTimeStep[ dtOneHour ];  interface  uses  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Chart, Series, ExtCtrls, Teengine, Buttons, TeeProcs;  TScrollForm = class(TForm) Chart1: TChart; LineSeries1: TLineSeries; Panel1: TPanel; Button1: TButton; BitBtn3: TBitBtn; CBVertical: TCheckBox; <> procedure LineSeries1AfterAdd(Sender: TChartSeries; ValueIndex: Longint); <> procedure		<u> </u>
ValueIndex: Longint); begin  { If VERTICAL SCROLL } if CBVertical.Checked then begin  With Sender.GetVertAxis do {< with the Vertical Axis}  Begin  Automatic := False; {< we dont want automatic scaling }  { In this example, we will set the Axis Minimum and Maximum values to  show One Hour of data ending at last point Time plus 5 minutes }  Minimum := 0;  Maximum := Sender.YValues.MaxValue + DateTimeStep[ dtFiveMinutes ];  Minimum := Maximum - DateTimeStep[ dtOneHour ];  interface  uses  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Chart, Series, ExtCtrls, Teengine, Buttons, TeeProcs;  LineSeries1: TLineSeries; Panel1: TPanel; Button1: TButton; BitBtn3: TBitBtn; CBVertical: TCheckBox; <> procedure LineSeries1AfterAdd(Sender: TChartSeries; ValueIndex: Longint); <> procedure	· ·	unit uscroll;
SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, With Sender.GetVertAxis do {		
If VERTICAL SCROLL     if CBVertical.Checked then     begin	ValueIndex: Longint);	interface
if CBVertical.Checked then begin  With Sender.GetVertAxis do {< with the Vertical Axis}  Begin  Automatic := False; {< we dont want automatic scaling }  { In this example, we will set the Axis Minimum and Maximum values to     show One Hour of data ending at last point Time plus 5 minutes }  Minimum := 0;  Maximum := Sender.YValues.MaxValue + DateTimeStep[ dtFiveMinutes ];  Minimum := Maximum - DateTimeStep[ dtOneHour ];  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,  Forms, Dialogs, StdCtrls, Chart, Series, ExtCtrls, Teengine, Buttons, TeeProcs;  LineSeries1: TChart; LineSeries1: TLineSeries; Button1: TButton; BitBtn3: TBitBtn; CBVertical: TCheckBox; > procedure LineSeries1AfterAdd(Sender: TChartSeries; ValueIndex: Longint);> procedure	begin	
begin  With Sender.GetVertAxis do {< with the Vertical Axis}  Begin  Automatic := False; {< we dont want automatic scaling }  { In this example, we will set the Axis Minimum and Maximum values to  show One Hour of data ending at last point Time plus 5 minutes }  Messages, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Chart, Series, ExtCtrls, Teengine, Buttons, TeeProcs;  TScrollForm = class(TForm) Chart1: TChart; LineSeries1: TLineSeries; Panel1: TPanel; Button1: TButton; BitBtn3: TBitBtn; CBVertical: TCheckBox; > procedure LineSeries1AfterAdd(Sender: TChartSeries; ValueIndex: Longint);> procedure	{ If VERTICAL SCROLL }	uses
begin  With Sender.GetVertAxis do {< with the Vertical Axis}  Begin  Automatic := False; {< we dont want automatic scaling }  { In this example, we will set the Axis Minimum and Maximum values to  show One Hour of data ending at last point Time plus 5 minutes }  Messages, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Chart, Series, ExtCtrls, Teengine, Buttons, TeeProcs;  type  TScrollForm = class(TForm) Chart1: TChart; LineSeries1: TLineSeries; Panel1: TPanel; Button1: TButton; BitBtn3: TBitBtn; CBVertical: TCheckBox; > procedure LineSeries1AfterAdd(Sender: TChartSeries; ValueIndex: Longint);> procedure	if CBVertical.Checked then	SysUtils, WinTypes, WinProcs,
With Sender.GetVertAxis do { < with the Vertical Axis }  Begin  Automatic := False; { < we dont want automatic scaling }  { In this example, we will set the Axis Minimum and Maximum values to	begin	¥
with the Vertical Axis }  Begin  Automatic := False; { < we dont want automatic scaling }  { In this example, we will set the Axis Minimum and Maximum values to	With Sender.GetVertAxis do { <	_
Begin Automatic := False; { < we dont want automatic scaling }  { In this example, we will set the Axis Minimum and Maximum values to     show One Hour of data ending at last point Time plus 5 minutes     }  Minimum := 0;  Maximum := Sender.YValues.MaxValue + DateTimeStep[ dtFiveMinutes ];  Minimum := Maximum - DateTimeStep[ dtOneHour ];  Series, ExtCtrls, Teengine, Buttons,  TeeProcs;  TeeProcsiteers  TeeProcsit	with the Vertical Axis }	Forms, Dialogs, StdCtrls, Chart,
Automatic := False; { < we dont want automatic scaling }  { In this example, we will set the Axis Minimum and Maximum values to	Begin	
dont want automatic scaling }  { In this example, we will set the Axis Minimum and Maximum values to  show One Hour of data ending at last point Time plus 5 minutes }  Minimum := 0;  Maximum := Sender.YValues.MaxValue + DateTimeStep[ dtFiveMinutes ];  Minimum := Maximum - DateTimeStep[ dtOneHour ];  type  TScrollForm = class(TForm)  Chart1: TChart;  LineSeries1: TLineSeries;  Panel1: TPanel;  Button1: TButton;  BitBtn3: TBitBtn;  CBVertical: TCheckBox;  <> procedure  LineSeries1AfterAdd(Sender: TChartSeries;  ValueIndex: Longint);  <> procedure	Automatic := False; { < we	
type  { In this example, we will set the Axis Minimum and Maximum values to	,	
{ In this example, we will set the Axis Minimum and Maximum values to  show One Hour of data ending at last point Time plus 5 minutes  }  Minimum := 0;  Maximum := Sender.YValues.MaxValue + DateTimeStep[ dtFiveMinutes ];  Minimum := Maximum - DateTimeStep[ dtOneHour ];  TScrollForm = class(TForm)  Chart1: TChart;  LineSeries1: TLineSeries;  Button1: TButton;  BitBtn3: TBitBtn;  CBVertical: TCheckBox;  <> procedure  LineSeries1AfterAdd(Sender: TChartSeries;  ValueIndex: Longint);  <> procedure		tyne
Axis Minimum and Maximum values to  show One Hour of data ending at last point Time plus 5 minutes  }  Minimum := 0;  Maximum :=  Sender.YValues.MaxValue +  DateTimeStep[ dtFiveMinutes ];  Minimum := Maximum -  DateTimeStep[ dtOneHour ];  Chart1: TChart;  LineSeries1: TLineSeries;  Button1: TButton;  BitBtn3: TBitBtn;  CBVertical: TCheckBox;  <> procedure  LineSeries1AfterAdd(Sender:  TChartSeries;  ValueIndex: Longint);  <> procedure	{ In this example, we will set the	
values toLineSeries1: TLineSeries;show One Hour of data ending at last point Time plus 5 minutesPanel1: TPanel;Button1: TButton;BitBtn3: TBitBtn;Maximum :=CBVertical: TCheckBox;Sender.YValues.MaxValue + DateTimeStep[ dtFiveMinutes ];LineSeries1AfterAdd(Sender: TChartSeries;Minimum := Maximum - DateTimeStep[ dtOneHour ];ValueIndex: Longint);		
show One Hour of data ending at last point Time plus 5 minutes  }  Minimum := 0;  Maximum := Sender.YValues.MaxValue + DateTimeStep[ dtFiveMinutes ];  Minimum := Maximum - DateTimeStep[ dtOneHour ];  DateTimeStep[ dtOneHour ];  Show One Hour of data ending Panel1: TPanel;  Button1: TButton;  BitBtn3: TBitBtn;  CBVertical: TCheckBox;  <> procedure  LineSeries1AfterAdd(Sender: TChartSeries;  ValueIndex: Longint);  <> procedure		,
at last point Time plus 5 minutes  }  Minimum := 0;  Maximum :=  Sender.YValues.MaxValue +  DateTimeStep[ dtFiveMinutes ];  Minimum := Maximum -  DateTimeStep[ dtOneHour ];  Button1: TButton;  BitBtn3: TBitBtn;  CBVertical: TCheckBox;  <> procedure  LineSeries1AfterAdd(Sender:  TChartSeries;  ValueIndex: Longint);  <> procedure		-
BitBtn3: TBitBtn;   Minimum := 0;   Maximum :=   Sender.YValues.MaxValue +     DateTimeStep[ dtFiveMinutes ];   Minimum := Maximum -     DateTimeStep[ dtOneHour ];   DateTimeStep[ dtOneHour ];		,
Minimum := 0;  Maximum := Sender.YValues.MaxValue + DateTimeStep[ dtFiveMinutes ];  Minimum := Maximum - DateTimeStep[ dtOneHour ];  CBVertical: TCheckBox;  <> procedure LineSeries1AfterAdd(Sender: TChartSeries;  ValueIndex: Longint);  <> procedure	}	
Maximum :=	Minimum := 0:	,
Sender.YValues.MaxValue + DateTimeStep[ dtFiveMinutes ];  Minimum := Maximum - DateTimeStep[ dtOneHour ];  LineSeries1AfterAdd(Sender: TChartSeries;  ValueIndex: Longint);  <> procedure		
DateTimeStep[ dtFiveMinutes ];  Minimum := Maximum - DateTimeStep[ dtOneHour ];  ThereStrict Add(Schder:  TChartSeries;  ValueIndex: Longint);  <> procedure		<u>-</u>
Minimum := Maximum -  DateTimeStep[ dtOneHour ];  ValueIndex: Longint);  <> procedure		· ·
DateTimeStep[ dtOneHour ]; <> procedure		,
procedure		
end; FormCreate(Sender: TObject);		<u> -</u>
	end;	FormCreate(Sender: TObject);

{ fill the LineSeries with some	end
random data }	else
LineSeries1.Clear; { < this	
	begin
removes all points from LineSeries1	With Sender.GetHorizAxis do {
}	< with the Horizontal Axis }
	Begin
{ let's add 60 minutes from 12:00 to	Automatic := False; { < we
12:59 }	dont want automatic scaling }
for t:= 0 to 59 do	
AddPoint( EncodeTime( 12, t,	{ In this example, we will set the
0,0),Random(100),clRed );	Axis Minimum and Maximum
	values to
{ let's add 60 more minutes from	show One Hour of data ending
13:00 to 13:59 }	at last point Time plus 5 minutes
for t:= 0 to 59 do	}
AddPoint( EncodeTime( 13, t,	Minimum := 0;
0,0),Random(100),clRed);	Maximum :=
end;	Sender.XValues.MaxValue +
,	DateTimeStep[ dtFiveMinutes ];
<> procedure	Minimum := Maximum -
TScrollForm.Button1Click(Sender:	DateTimeStep[ dtOneHour ];
TObject);	end;
var h,m,s,msec:word;	End;
begin	
	end;
if CBVertical.Checked then { if	
VERTICAL SCROLL }	<> procedure
DecodeTime(	TScrollForm.AddPoint(Const
LineSeries1.YValues.Last, h, m, s,	x,y:Double; AColor:TColor);
msec)	begin
else	if CBVertical.Checked then { If
DecodeTime(	VERTICAL SCROLL }
LineSeries1.XValues.Last, h, m, s,	
msec);	LineSeries1.AddXY(y,x,",AColor)
	else
{ add a new random point to the	
Series (one more minute) }	LineSeries1.AddXY(x,y,'',AColor);
inc(m);	end;
if m=60 then	
begin	<> procedure
m:=0;	TScrollForm.FormCreate(Sender:
inc(h);	TObject);
end;	begin
AddPoint( EncodeTime( h, m, s,	FillDemoPoints;
msec), Random(100), clYellow);	end;
end;	
	<> procedure
<> procedure	TScrollForm.FillDemoPoints;
TScrollForm.CBVerticalClick(Sende	
The state of the s	var t:Longint;
r: TObject);	begin

EvtCtula
ExtCtrls;
tyna
type TSeriesForm = class(TForm)
GroupBox1: TGroupBox;
BitBtn4: TBitBtn;
BitBtn5: TBitBtn;
BitBtn6: TBitBtn;
BitBtn7: TBitBtn;
GroupBox3: TGroupBox;
BitBtn9: TBitBtn;
BitBtn10: TBitBtn;
BitBtn11: TBitBtn;
BStacked: TBitBtn;
BFastLine: TBitBtn;
BitBtn14: TBitBtn;
BArrowSeries: TBitBtn;
BitBtn1: TBitBtn;
BStackedAreas: TBitBtn;
BitBtn21: TBitBtn;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
Label8: TLabel;
Label9: TLabel;
Panel1: TPanel;
Panel2: TPanel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
<> procedure
BitBtn4Click(Sender: TObject);
<> procedure
BitBtn5Click(Sender: TObject);
<> procedure
BitBtn6Click(Sender: TObject);
<> procedure
BitBtn7Click(Sender: TObject);
<> procedure
BitBtn9Click(Sender: TObject);
<pre>&lt;&gt; procedure PitPtn10Click(Sondor: TObject);</pre>
BitBtn10Click(Sender: TObject);

begin
With LineSeries1 do
if CBVertical.Checked then { If
VERTICAL SCROLL }
begin
0
YValues.Order:=loAscending;
XValues.Order:=loNone;
end
else
begin
XValues.Order:=loAscending;
YValues.Order:=loNone;
end;
Chart1.LeftAxis.Automatic:=True;
{ < this makes axis scales
AUTOMATIC AGAIN!}
Chart1.BottomAxis.Automatic:=Tru
e; { < this makes axis scales
AUTOMATIC AGAIN!}
FillDemoPoints; { < fill sample
values again!}
end;
cnu,
,
end.
111111111111111111111111111111111111111
36
{***********
*******
TaoChart Dolnhi Companant
{ TeeChart Delphi Component
Library }
{ Main Form Demo
}
{ Copyright (c) 1995-2001 by David
Berneda }
{ All rights reserved }
{**********
{**********************
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
{**********************
{************ ************ unit udemocha;
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
{************ ************ unit udemocha;
{************ ************ unit udemocha;
{*************************************
{*************************************
{*************************************
{*************************************

Free;	<> procedure
end;	BitBtn11Click(Sender: TObject);
End;	<> procedure
	FormCreate(Sender: TObject);
<> procedure	<> procedure
TSeriesForm.BitBtn4Click(Sender:	BStackedClick(Sender: TObject);
TObject);	<> procedure
begin	BFastLineClick(Sender: TObject);
ShowForm(TBasicForm);	<> procedure
end;	BitBtn14Click(Sender: TObject);
	<> procedure
<> procedure	BArrowSeriesClick(Sender:
TSeriesForm.BitBtn5Click(Sender:	TObject);
TObject);	<> procedure
begin	BStackedAreasClick(Sender:
ShowForm(TPieForm);	TObject);
end;	<> procedure
	BitBtn21Click(Sender: TObject);
<> procedure	private
TSeriesForm.BitBtn6Click(Sender:	{ Private declarations }
TObject);	public
begin	{ Public declarations }
ShowForm(TBubbleForm);	<> procedure
end;	ShowForm(AFormClass:TFormClas
,	s);
<> procedure	end;
TSeriesForm.BitBtn7Click(Sender:	
TObject);	implementation
begin	
ShowForm(TGanttForm);	{\$R *.dfm}
end;	Uses
	Pie,Bubble,Gantt,Basic,UStack,UFas
<> procedure ShowNeedsBDE;	<b>t</b> ,
Begin	UArrows,StackAre,UHighLo
ShowMessage('Needs the Borland	<b>{\$IFNDEF NODB}</b>
Database Engine.'+#13+#10+	
'Please email to get the full	,TablePie,SQLBars,Linked,UDBHor
TeeChart DEMO.');	iz
end;	{\$ENDIF}
	;
<> procedure	
TSeriesForm.BitBtn9Click(Sender:	<> procedure
TObject);	TSeriesForm.ShowForm(AFormCla
begin	ss:TFormClass);
{\$IFNDEF NODB}	Begin
ShowForm(TTablePieForm);	With AFormClass.Create(Self) do
{\$ELSE}	try
ShowNeedsBDE;	ShowModal;
{\$ENDIF}	finally

	•
TSeriesForm.BFastLineClick(Sende	end;
r: TObject);	
begin	<> procedure
ShowForm(TFastLineForm);	TSeriesForm.BitBtn10Click(Sender:
end;	TObject);
	begin
<> procedure	{\$IFNDEF NODB}
TSeriesForm.BitBtn14Click(Sender:	ShowForm(TSQLBarsForm);
TObject);	{\$ELSE}
begin	ShowNeedsBDE;
{\$IFNDEF NODB}	{\$ENDIF}
ShowForm(TDBHorizBarForm);	end;
{\$ELSE}	
ShowNeedsBDE;	<> procedure
{\$ENDIF}	TSeriesForm.BitBtn11Click(Sender:
end;	TObject);
	begin
/ > nwaaadura	{\$IFNDEF NODB}
<> procedure TSoviesForm PAymont Sovies Click (So	
TSeriesForm.BArrowSeriesClick(Se	ShowForm(TLinkedTablesForm);
nder: TObject);	{\$ELSE}
begin	ShowNeedsBDE;
ShowForm(TArrowsForm);	{\$ENDIF}
end;	end;
<> procedure	<> procedure
TSeriesForm.BStackedAreasClick(S	TSeriesForm.FormCreate(Sender:
ender: TObject);	TObject);
begin	begin
ShowForm(TAreasForm);	if Screen.Width<800 then
end;	ShowMessage('Warning:
	'+#13+#10+
<> procedure	'This Demo is best viewed
TSeriesForm.BitBtn21Click(Sender:	with a Screen'+#13+#10+
TObject);	'resolution of 800x600 or
begin	greater,'+#13+#10+
ShowForm(THighLowForm);	'and a Color Depth of 256
end;	or greater.'+#13+#10+
, , , , , , , , , , , , , , , , , , ,	'16K Colors is also better
end.	than 256 Colors.');
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	end;
	citu,
37 	/ > nuocoduno
l f	<pre>&lt;&gt; procedure TS&gt; PS4 PS4</pre>
******	TSeriesForm.BStackedClick(Sender
{ TeeChart. TChart Component	: TObject);
}	begin
{ Copyright (c) 1995-2001 by David	ShowForm(TStackedForm);
Berneda }	end;
{ All Rights Reserved }	
{*********	<> procedure
	procedure

[move as
TObject);
begin
With LineSeries1 do
Begin
XValues.DateTime:=False;
Clear;
Add( 100, 'Jan', clTeeColor );
Add( 140, 'Feb', clTeeColor );
Add( 120, 'Mar', clTeeColor );
Add( 110, 'Apr', clTeeColor );
Add( 80, 'May', clTeeColor );
Add( 90, 'Jun', clTeeColor );
Add( 75, 'Jul', clTeeColor );
Add( 85, 'Aug', clTeeColor );
Add( 95, 'Sep', clTeeColor );
Add( 105, 'Oct', clTeeColor );
Add(100, 'Nov', clTeeColor);
Add( 90, 'Dec', clTeeColor );
end;
With LineSeries2 do
Begin
XValues.DateTime:=False;
Clear;
Add( 60, 'Jan', clTeeColor );
Add( 100, 'Feb', clTeeColor );
Add( 20, 'Mar', clTeeColor );
Add( 70, 'Apr', clTeeColor );
AreaBrush:=bsCross;
end;
With ChartShape1 do
begin
X0:= 2;
X1:= 8;
Y0:= 100;
Y1:= 130;
end;
With ChartShape2 do
begin
X0:= 5;
X1:= 9;
Y0:= 60;
Y1:= 80;
end;
With ChartShape3 do
begin
X0:= 1;
X1:= 4;
· · · · · · · · · · · · · · · · · · ·
Y0:= 135;

******
unit ushapes;
unit usimpes,
interface
uses
SysUtils, WinTypes, WinProcs,
Messages, Classes, Graphics,
Controls,
Forms, Dialogs, TeeShape, Chart,
Series, ExtCtrls, StdCtrls, Teengine,
Buttons, TeeProcs;
type
TShapesForm = class(TForm)
Chart1: TChart;
LineSeries1: TLineSeries;
ChartShape1: TChartShape;
LineSeries2: TLineSeries;
ChartShape2: TChartShape;
Panel1: TPanel;
Label1: TLabel;
Label2: TLabel;
ChartShape3: TChartShape;
Panel2: TPanel;
BitBtn2: TBitBtn;
Memo1: TMemo;
<> procedure
FormCreate(Sender: TObject);
<> procedure
ChartShape3Click(Sender:
TChartSeries; ValueIndex: Integer;
Button: TMouseButton; Shift:
TShiftState; X, Y: Integer);
private
{ Private declarations }
public
{ Public declarations }
end;
Characteristic TCharacteristic
ShapesForm: TShapesForm;
implementation
implementation
(CD * dfm)
{\$R *.dfm}
Z Nugas durus
<> procedure TShanas Form Form Croata(Sandar:
TShapesForm.FormCreate(Sender:

# <u>AC.&P</u>

D + G 1 TD + G	¥74 400
DataSource1: TDataSource;	Y1:= 120;
Panel1: TPanel;	end;
BitBtn1: TBitBtn;	end;
BarSeries1: TBarSeries;	
Query1: TQuery;	<> procedure
Panel2: TPanel;	TShapesForm.ChartShape3Click(Se
Memo1: TMemo;	nder: TChartSeries;
BitBtn2: TBitBtn;	ValueIndex: Integer; Button:
ComboBox1: TComboBox;	TMouseButton; Shift: TShiftState;
Label1: TLabel;	X,
CBRandomBar: TCheckBox;	Y: Integer);
Panel3: TPanel;	begin
DBGrid1: TDBGrid;	ShowMessage('You clicked the
Label2: TLabel;	ChartShape3 (Rectangle)');
<> procedure	end;
BitBtn2Click(Sender: TObject);	
<> procedure	end.
ComboBox1Change(Sender:	
TObject);	
<> procedure	20
FormCreate(Sender: TObject);	38
<> procedure	<pre>{************************************</pre>
DBChart1ClickLegend(Sender:	*********
TCustomChart;	<u> </u>
Button: TMouseButton; Shift:	{ TeeChart Delphi Component
TShiftState; X, Y: Integer);	Library } { SQL Query Chart Demo
<> procedure BarSeries1Click(Sender:	SQL Query Chart Demo
TChartSeries; ValueIndex: Integer;	{ Copyright (c) 1995-2001 by David
Button: TMouseButton; Shift:	Berneda }
TShiftState; X, Y: Integer);	{ All rights reserved }
<> procedure	{***********
BarSeries1GetBarStyle(Sender:	**************************************
TCustomBarSeries;	unit sqlbars;
ValueIndex: Longint; var	unit squars,
TheBarStyle: TBarStyle);	interface
<> procedure	
CBRandomBarClick(Sender:	uses
TObject);	SysUtils, WinTypes, WinProcs,
private	Messages, Classes, Graphics,
{ Private declarations }	Controls,
public	Forms, Dialogs, StdCtrls, Buttons,
{ Public declarations }	ExtCtrls, Grids, DBGrids, DB,
end;	DBTables, Chart, Series, DbChart,
	Teengine, TeeProcs;
implementation	
	type
{\$R *.dfm}	TSQLBarsForm = class(TForm)
Uses UDemUtil;	DBChart1: TDBChart;
,	,

1 TOL 10	
nder: TChartSeries;	
ValueIndex: Integer; Button:	<> procedure
TMouseButton; Shift: TShiftState;	TSQLBarsForm.BitBtn2Click(Sende
X,	r: TObject);
Y: Integer);	begin { rerun the SQL query }
begin	Screen.Cursor:=crHourGlass;
With BarSeries1 do	try
	Query1.Close;
ValueColor[ValueIndex]:=EditColor	Query1.Sql:=Memo1.Lines;
(Self,ValueColor[ValueIndex]);	Query1.Open;
end;	finally
	Screen.Cursor:=crDefault;
<> procedure	end;
TSQLBarsForm.BarSeries1GetBarS	end;
tyle(Sender: TCustomBarSeries;	
ValueIndex: Longint; var	<> procedure
TheBarStyle: TBarStyle);	TSQLBarsForm.ComboBox1Chang
begin	e(Sender: TObject);
if CBRandomBar.Checked then	begin
	if BarSeries1 is TCustomBarSeries
TheBarStyle:=TBarStyle(Random(1	then
+Ord(High(TBarStyle))));	
end;	BarSeries1.BarStyle:=TBarStyle(Co
	mboBox1.ItemIndex); { < change
<> procedure	bar style }
TSQLBarsForm.CBRandomBarClic	end;
k(Sender: TObject);	,
begin	<> procedure
ComboBox1.Enabled:=not	TSQLBarsForm.FormCreate(Sende
CBRandomBar.Checked;	r: TObject);
DBChart1.Repaint;	begin
end;	9
,	ComboBox1.ItemIndex:=Ord(BarSe
end.	ries1.BarStyle); { < set combobox1
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	<b> </b> }
38	end;
{**********	,
********	<> procedure
{ TeeChart Delphi Component	TSQLBarsForm.DBChart1ClickLeg
Library }	end(Sender: TCustomChart;
{ Stacked Bar Series Example	<b>Button: TMouseButton; Shift:</b>
}	TShiftState; X, Y: Integer);
{ Copyright (c) 1995-2001 by David	begin
Berneda }	With DBChart1.Legend do
{ All rights reserved }	Color:=EditColor(Self,Color);
{**********	end;
**********	
unit ustack;	<> procedure
	TSQLBarsForm.BarSeries1Click(Se
	¿ Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z

Integer);	interface
<> procedure	meriace
CheckBox2Click(Sender: TObject);	uses
private	SysUtils, WinTypes, WinProcs,
{ Private declarations }	Messages, Classes, Graphics,
public	Controls,
{ Public declarations }	Forms, Dialogs, Chart, Series,
	ExtCtrls, StdCtrls, Teengine,
<> procedure RefreshShape;	ArrowCha,
end;	Buttons, teeprocs;
implementation	Buttons, teeproes,
implementation	type
(CD * dfm)	TStackedForm = class(TForm)
{\$R *.dfm}	Chart1: TChart;
Uses UDemUtil,tecanvas;	BarSeries1: TBarSeries;
< > nwoooduwo	BarSeries2: TBarSeries;
<> procedure TStackedForm FormCreate(Sondon)	RadioGroup1: TRadioGroup;
TStackedForm.FormCreate(Sender:	Panel1: TPanel;
TObject);	,
var t:Longint;	Button1: TButton;
begin	RadioGroup2: TRadioGroup;
{ The Chart gradient is visible only	ComboBox1: TComboBox;
with 16k Colors or greater video	Label1: TLabel;
mode }	CheckBox1: TCheckBox;
	Timer1: TTimer;
Chart1.Gradient.Visible:=Chart1.Is	BarSeries3: TBarSeries;
ScreenHighColor;	Shape1: TShape;
	BitBtn3: TBitBtn;
{ Add some random points to the	CheckBox2: TCheckBox;
three bar Series }	<> procedure
Randomize;	FormCreate(Sender: TObject);
with BarSeries1 do	<> procedure
for t:=1 to 12 do Add(	RadioGroup1Click(Sender:
Random(100),ShortMonthNames[t],	TObject);
clTeeColor);	<> procedure
with BarSeries2 do	Button1Click(Sender: TObject);
for t:=1 to 12 do Add(	<> procedure
Random(100),ShortMonthNames[t],	ComboBox1Change(Sender:
clTeeColor); with BarSeries3 do	TObject);
	<> procedure
for t:=1 to 12 do Add(	RadioGroup2Click(Sender:
Random(100),ShortMonthNames[t],	TObject);
clTeeColor);	<pre>&lt;&gt; procedure Chask Pay 1 Click (Sanday TObicat);</pre>
(Fill a Comba Day with the three	CheckBox1Click(Sender: TObject);
{ Fill a ComboBox with the three	<> procedure
Series titles }	Timer1Timer(Sender: TObject);
ComboBox1.Items.Clear;	<> procedure
for t:=0 to Chart1.SeriesCount-1 do	Shape1MouseUp(Sender: TObject;
	Button: TMouseButton;
ComboBox1.Items.Add(Chart1.Serie	Shift: TShiftState; X, Y:

	-[4] N)
marks }	s[t].Name);
SeriesDown(Series[0]);	ComboBox1.ItemIndex:=0;
{ move series 0 to back }	
Series[SeriesCount-	RefreshShape;
1].Marks.Visible:=True; { show	end;
marks again }	
end;	<> procedure
ComboBox1Change(Self);	TStackedForm.RefreshShape;
end;	Begin
,	{ This method sets the Shape color
<> procedure	to the active Series color in
TStackedForm.ComboBox1Change(	Combobox }
Sender: TObject);	With
begin	Chart1.Series[ComboBox1.ItemInde
{ Change the active Bar Series }	x do
With Chart 1 Savias   Camba Pay 1   Itam Inda	Begin
Chart1.Series[ComboBox1.ItemInde	Shape1.Visible:=not
x] as TBarSeries do	ColorEachPoint;
	if Shape1. Visible then
RadioGroup2.Itemindex:=Ord(BarS	Shape1.Brush.Color:=SeriesColor;
tyle);	end;
RefreshShape;	end;
end;	
	<> procedure
<> procedure	TStackedForm.RadioGroup1Click(S
TStackedForm.RadioGroup2Click(S	ender: TObject);
ender: TObject);	begin
begin	{ Four ways to plot bar Series: }
{ Change the active Bar Series Style	Case RadioGroup1.ItemIndex of
}	0:
With	BarSeries1.MultiBar:=mbNone;
Chart1.Series[ComboBox1.ItemInde	1: BarSeries1.MultiBar:=mbSide;
x] as TBarSeries do	2:
A) us i bui series uo	
BarStyle:=TBarStyle(RadioGroup2.	BarSeries1.MultiBar:=mbStacked; 3:
Itemindex);	
	BarSeries1.MultiBar:=mbStacked10
end;	0;
Z	end;
<> procedure	end;
TStackedForm.CheckBox1Click(Sen	
der: TObject);	<> procedure
begin	TStackedForm.Button1Click(Sender
{ Start / Stop Animation }	: TObject);
	begin
Timer1.Enabled:=CheckBox1.Check	{ Scroll two Bar Series }
ed;	With Chart1 do
RadioGroup1.Enabled:=not	Begin
Timer1.Enabled;	Series[SeriesCount-
RadioGroup2.Enabled:=not	1].Marks.Visible:=False; { hide
	Time in the second of the second seco

end;	Timer1.Enabled;
Citu,	if not Timer 1. Enabled then
{ this function changes randomly	ComboBox1Change(Self);
the Brush parameter }	end;
	enu,
<> procedure	
RandomBrush(ABrush:TBrush);	<> procedure
Begin	TStackedForm.Timer1Timer(Sende
With ABrush do	r: TObject);
Case Random(2) of	
0: Color:=RandomColor(True);	{ this function returns a random
1: if Random(10)<5 then	color from ColorPalette }
Style:=TBrushStyle(Random(8));	Function
end;	RandomColor(CheckBack:Boolean):
end;	TColor;
	Begin
{ this function changes randomly	Repeat
the Font parameter }	
<> procedure	result:=ColorPalette[1+Random(Ma
RandomFont(AFont:TFont);	xDefaultColors)];
Begin	Until (not CheckBack) or
With AFont do	( (result<>Chart1.Color) and
Case Random(2) of	(result<>Chart1.BackColor));
0: Color:=RandomColor(True);	end;
` ' '	
1: if Random(2)=0 then	{ this function returns a random
Size:=Size+1	angle from 0, 90, 180 or 270 }
else	Function RandomAngle:Integer;
if Size>7 then Size:=Size-1;	
end;	Begin  Casa Bandam(4) of
End;	Case Random(4) of
	0: result:=0;
var tmpSeries:TBarSeries;	1: result:=90;
t:Longint;	2: result:=180;
begin	else
{ This long routine	{ 3: } result:=270;
is only a random generator for	end;
Chart and Series properties.	end;
}	
Timer1.Enabled:=False; { stop	{ this function changes randomly
timer }	the Pen parameter }
	<> procedure
{ Set all Series.Active }	RandomPen(APen:TChartPen);
for t:=0 to Chart1.SeriesCount-1 do	Begin
Chart1.Series[t].Active:=True;	With APen do
	if Visible then
{ Choose a Series randomly }	Case Random(3) of
( Choose a Series Landonny )	0: Color:=RandomColor(True);
tmpSeries:=Chart1.Series[Random(	1: Style:=TPenStyle(Random(5));
- `	2: Width:=1+Random(3);
ComboBox1.Items.Count)] as TBarSeries;	
1 Dar Series;	end;

Then, lets change this chart a little	RandomFont(Chart1.LeftAxis.Title.	
19: RandomFont(Chart1.BottomAxis.Ti tle.Font); 20: tmpSeries.Marks.Style:=TSeriesMar ksStyle(Random(9)); 21: With Chart1 do Begin BevelWidth:=1+Random(10); MarginTop :=TeeDefVerticalMargin+BevelWidth; MarginRight :=TeeDefHorizMargin+BevelWidth; MarginBottom:=TeeDefVerticalMargin+BevelWidth; cnd; 22: Chart1.Legend.Visible:=not Chart1.Legend.Visible:=sbserie s else Chart1.Legend.LegendStyle:=lsSerie s Chart1.Legend.LegendStyle:=lsValue es; 25: tmpSeries.Marks.Visible; 25: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible; 25: tmpSeries.Marks.Visible; 26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible; 27: RandomPen(Chart1.BottomAxis.Grid); 29: Chart1.Legend.Frame.Visible:=not	`	S Than late change this chart a
RandomFont(Chart1.BottomAxis.Title.Font);  20: tmpSeries.Marks.Style:=TSeriesMarksStyle(Random(9));  21: With Chart1 do Begin BevelWidth:=1+Random(10); MarginTop :=TeeDefUerticalMargin+BevelWidth; MarginLeft :=TeeDefHorizMargin+BevelWidth; MarginBottom:=TeeDefVerticalMargin+BevelWidth; MarginBottom:=TeeDefVerticalMargin+BevelWidth; end; 22: Chart1.Legend.Visible:=not Chart1.Legend.LegendStyle:=IsSeries  Chart1.Legend.LegendStyle:=IsSeries  Chart1.Legend.LegendStyle:=IsSeries  Chart1.Legend.LegendStyle:=IsSeries  Chart1.Legend.LegendStyle:=IsSeries  Chart1.Legend.LegendStyle:=IsSeries  Chart1.Legend.Alignment:=IaRight  11: Chart1.Legend.Alignment:=IaRight  Chart1.Legend.Alignment:=IaRight;  11: Chart1.Legend.Ali	,,	,
tle.Font); 20: tmpSeries.Marks.Style:=TSeriesMarksStyle(Random(9)); 21: With Chart1 do Begin BevelWidth:=HRandom(10); MarginTop :=TeeDefVerticalMargin+BevelWidth; MarginRight :=TeeDefHorizMargin+BevelWidth; MarginBottom:=TeeDefVerticalMargin+BevelWidth; end; 22: Chart1.Legend.Visible:=not Chart1.Legend.Visible; 23: if Random(10)<5 then Chart1.Legend.LegendStyle:=IsSerie s Clart (Lagend.LegendStyle:=IsSerie s Clart1.Legend.LegendStyle:=IsSerie s Else Chart1.Legend.LegendStyle:=IsSerie s Clart1.Legend.LegendStyle:=IsSerie s Clart1.Legend.LegendStyle:=IsSerie s RandomPen(Chart1.BottomAxis.Grid); 26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible:=not thart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAx		,
20: tmpSeries.Marks.Style:=TSeriesMarksStyle(Random(9)): 21: With Chart1 do Begin BevelWidth:=1+Random(10): MarginTop :=TeeDefVerticalMargin+BevelWidth; MarginLeft :=TeeDeffIlorizMargin+BevelWidth; MarginRight :=TeeDeffIlorizMargin+BevelWidth; MarginBottom:=TeeDefVerticalMargin+BevelWidth; cnd; 22: Chart1.Legend.Visible:=not Chart1.Legend.Visible: 23: if Random(10)<5 then Chart1.Legend.LegendStyle:=IsSerie s else Chart1.Legend.LegendStyle:=IsValu es; 24: tmpSeries.Active: 25: tmpSeries.Active: 26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible: 27: RandomPen(Chart1.BottomAxis.Grid); 28: RandomPen(Chart1.BottomAxis.Grid); 29: Chart1.Legend.Frame.Visible: 30: RadioGroup1.ItemIndex:=Random(RadioGroup2.Items.Count); 6: Button1Click(Self); 7: tmpSeries.Series Color:=RandomColor(False); 8: Chart1.Chart3dPercent:=5+Random(80); 9: Chart1.Legend.Alignment=laRight then Chart1.Legend.Alignment:=laRight; 11: RandomFont(Chart1.BottomAxis.LabelsFont); 12: RandomFont(Chart1.BottomAxis.LabelsFont); 13: Chart1.View3d:=not Chart1.View3d: 14: tmpSeries.BarWidthPercent:=50+R andom(40); 15: RandomFont(tmpSeries.Marks.Font); 16: Chart1.BottomAxis.Grid.Visible:=not Chart1.Legend.Frame.Visible: 17: RandomFont(tmpSeries.Marks.Font); 16: Chart1.BottomAxis.Grid.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.LeftAxis.Grid.Visible:=not	` `	` /
tmpSeries.Marks.Style:=TSeriesMarksStyle(Random(9)); 21: With Chart1 do Begin BevelWidth:=1+Random(10); MarginTop :=TeeDefVerticalMargin+BevelWidth; MarginRight :=TeeDefIlorizMargin+BevelWidth; MarginBibttom:=TeeDefVerticalMargin+BevelWidth; Cend; 22: Chart1.Legend.Visible:=not Chart1.Legend.Visible: 23: if Random(10)<5 then Chart1.Legend.LegendStyle:=IsValues; 24: tmpSeries.Active:=not tmpSeries.Active:=not tmpSeries.Marks.BackColor:=RandomColor(True); 25: tmpSeries.Marks.Visible:=not cmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible:=not chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.	*	
Lage of Legend Style:   Legend Legend Legend Style:   Legend Legend Legend Style:   Legend Legen		_ ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` `
21: With Chart1 do Begin BevelWidth:=1+Random(10); MarginTop :=TeeDefVerticalMargin+BevelWidth h; MarginLeft :=TeeDefHorizMargin+BevelWidth; MarginBit :=TeeDefHorizMargin+BevelWidth; MarginBottom:=TeeDefVerticalMargin+BevelWidth; end; 22: Chart1.Legend.Visible:=not Chart1.Legend.Visible; 23: if Random(10)<5 then Chart1.Legend.LegendStyle:=lsSerie s else Chart1.Legend.Alignment:=laRight then  Chart1.Legend.Alignment:=laRight; 11: RandomFont(Chart1.BottomAxis.La belsFont); 12: RandomFont(Chart1.BottomAxis.La belsFont); 12: RandomFont(Chart1.LeftAxis.Label sFont); 13: Chart1.View3d:=not Chart1.View3d: 14: tmpSeries.BarWidthPercent:=50+R andom(40); 15: RandomPen(Chart1.LeftAxis.Grid); 29: Chart1.Legend.Frame.Visible:=not Chart1.BottomAxis.Grid.Visible:=not thart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Gri	=   · · · · · · · · · · · · · · · · ·	1 /
Begin BevelWidth:=1+Random(10); MarginTop :=TeeDefVerticalMargin+BevelWidth; MarginLeft :=TeeDefHorizMargin+BevelWidth; MarginRight :=TeeDefHorizMargin+BevelWidth; MarginBottom:=TeeDefVerticalMargin+BevelWidth; end; 22: Chart1.Legend.Visible:=not Chart1.Legend.Visible; 23: if Random(10)<5 then Chart1.Legend.Visible; 23: if Random(10)<5 then Chart1.Legend.LegendStyle:=lsSerie s else Chart1.Legend.Alignment:=laRight then  Chart1.Legend.Alignment:=laRight then  Chart1.Legend.Alignment:=laRight; 11: RandomFont(Chart1.BottomAxis.LabelsFont); 12: RandomFont(Chart1.BottomAxis.LabelsFont); 13: Chart1.View3d: 14: tmpSeries.Marks.Visible; 27: RandomPen(Chart1.BottomAxis.Grid.); 29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible; 30: RandomPen(Chart1.Legend.Frame); 16: Chart1.BottomAxis.Grid.Visible:=not Chart1.BottomAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not C		
BevelWidth:=1+Random(10); MarginTop :=TeeDefVerticalMargin+BevelWidth; MarginLeft :=TeeDefHorizMargin+BevelWidth; MarginBipt :=TeeDefHorizMargin+BevelWidth; MarginBottom:=TeeDefVerticalMargin+BevelWidth; end; 22: Chart1.Legend.Visible:=not Chart1.Legend.Visible; 23: if Random(10)<5 then Chart1.Legend.LegendStyle:=IsSerie s else Chart1.Legend.LegendStyle:=IsSerie s else Chart1.Legend.LegendStyle:=IsValues; 24: tmpSeries.Active:=not tmpSeries.Active; 25: tmpSeries.Marks.BackColor:=RandomColor(Chart1.Legend.Alignment:=laRight; 11: RandomFont(Chart1.BottomAxis.LabelsFont); 12: RandomFont(Chart1.BottomAxis.LabelsFont); 13: Chart1.View3d:=not Chart1.View3d: 14: tmpSeries.BarWidthPercent:=50+R andom(40); 15: RandomFont(tmpSeries.Marks.Font); 15: RandomFont(tmpSeries.Marks.Font); 16: Chart1.BottomAxis.Grid.Visible:=not Chart1.Legend.Frame.Visible; 30: RandomPen(Chart1.Legend.Frame); 17: Chart1.BottomAxis.Grid.Visible:=not Chart1.Legend.Alignment:=laRight; 11: RandomFont(Chart1.BottomAxis.LabelsFont); 12: RandomFont(Chart1.LeftAxis.Crid); 14: tmpSeries.SarWidthPercent:=50+R andom(40); 15: RandomFont(tmpSeries.Marks.Font); 16: Chart1.BottomAxis.Grid.Visible:=not Chart1.Legend.Visible;=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible;=not Chart1.LeftAxis.Grid.Visible;		
MarginTop :=TeeDefVerticalMargin+BevelWidth; MarginLeft :=TeeDefHorizMargin+BevelWidth; MarginBottom:=TeeDefVerticalMargin+BevelWidth; MarginBottom:=TeeDefVerticalMargin+BevelWidth; end; 22: Chart1.Legend.Visible:=not Chart1.Legend.Visible; 23: if Random(10)<5 then Chart1.Legend.LegendStyle:=IsSeries else Chart1.Legend.LegendStyle:=IsSeries else Chart1.Legend.Alignment=IaRight then  Chart1.Legend.Alignment:=IaLeft else  Chart1.Legend.Alignment:=IaRight; 11: RandomFont(Chart1.BottomAxis.LabelsFont); 12: RandomFont(Chart1.BottomAxis.LabelsFont); 13: Chart1.View3d:=not Chart1.View3d:=not Chart1.View3d: 14: tmpSeries.BarWidthPercent:=50+R andom(40); 15: RandomPen(Chart1.LeftAxis.Grid); 29: Chart1.Legend.Frame.Visible:=not Chart1.BottomAxis.Grid.Visible:=not Chart1.Legend.Frame); 16: Chart1.BottomAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible;=not Chart1.LeftAxis.Grid.Visible;=	Begin	RadioGroup2.Items.Count);
:=TeeDefVerticalMargin+BevelWidth;  MarginLeft :=TeeDefHorizMargin+BevelWidth; MarginBottom:=TeeDefVerticalMargin+BevelWidth;  MarginBottom:=TeeDefVerticalMargin+BevelWidth; end; 22: Chart1.Legend.Visible:=not Chart1.Legend.Visible:=105 Chart1.Legend.Visible:=105 Chart1.Legend.LegendStyle:=IsSerie s  else Chart1.Legend.Alignment=IaRight then  Chart1.Legend.Alignment:=IaLeft else  Chart1.Legend.Alignment:=IaLeft else  Chart1.Legend.Alignment:=IaRight; 11: RandomFont(Chart1.BottomAxis.LabelsFont); 12: RandomFont(Chart1.LeftAxis.LabelsFont); 12: RandomFont(Chart1.LeftAxis.LabelsFont); 13: Chart1.View3d;=not Chart1.View3d; 14: tmpSeries.BarWidthPercent:=50+R andom(40); 15: RandomFont(tmpSeries.Marks.Font); 16: Chart1.BottomAxis.Grid.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1	BevelWidth:=1+Random(10);	6: Button1Click(Self);
h;  MarginLeft :=TeeDefHorizMargin+BevelWidth; MarginBottom:=TeeDefVerticalMargin+BevelWidth; end; 22: Chart1.Legend.Visible:=not Chart1.Legend.Visible; 23: if Random(10)<5 then Chart1.Legend.LegendStyle:=lsSerie s else Chart1.Legend.LegendStyle:=lsValu es; 24: tmpSeries.Active: 25: tmpSeries.Active; 25: tmpSeries.Marks.BackColor:=Rand omColor(True); 26: tmpSeries.Marks.Visible; 27: RandomPen(Chart1.BottomAxis.Grid); 29: Chart1.Legend.Frame.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible;=not	MarginTop	7:
MarginLeft :=TeeDefHorizMargin+BevelWidth; MarginBottom:=TeeDefVerticalMargin+BevelWidth;  MarginBottom:=TeeDefVerticalMargin+BevelWidth; end; 22: Chart1.Legend.Visible:=not Chart1.Legend.Visible:=not Chart1.Legend.Visible:=sSerie s  clse Chart1.Legend.Alignment=laRight then  Chart1.Legend.Alignment:=laLeft else  Chart1.Legend.Alignment:=laLeft else  Chart1.Legend.Alignment:=laRight; 11: RandomFont(Chart1.BottomAxis.LabelsFont); 12: RandomFont(Chart1.BottomAxis.LabelsFont); 12: RandomPen(Chart1.BottomAxis.Grid); 28: RandomPen(Chart1.LeftAxis.Grid); 29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Alignment:=laRight then  Chart1.Legend.Alignment:=laRight; 11: RandomFont(Chart1.BottomAxis.LabelsFont); 12: RandomFont(Chart1.LeftAxis.LabelsFont); 13: Chart1.View3d:=not Chart1.View3d:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Alignment:=laRight then  Chart1.Legend.Alignment:=laRigh	:=TeeDefVerticalMargin+BevelWidt	tmpSeries.SeriesColor:=RandomCol
MarginLeft :=TeeDefHorizMargin+BevelWidth; MarginBottom:=TeeDefVerticalMargin+BevelWidth;  MarginBottom:=TeeDefVerticalMargin+BevelWidth; end; 22: Chart1.Legend.Visible:=not Chart1.Legend.Visible:=not Chart1.Legend.Visible:=sSerie s  clse Chart1.Legend.Alignment=laRight then  Chart1.Legend.Alignment:=laLeft else  Chart1.Legend.Alignment:=laLeft else  Chart1.Legend.Alignment:=laRight; 11: RandomFont(Chart1.BottomAxis.LabelsFont); 12: RandomFont(Chart1.BottomAxis.LabelsFont); 12: RandomPen(Chart1.BottomAxis.Grid); 28: RandomPen(Chart1.LeftAxis.Grid); 29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Alignment:=laRight then  Chart1.Legend.Alignment:=laRight; 11: RandomFont(Chart1.BottomAxis.LabelsFont); 12: RandomFont(Chart1.LeftAxis.LabelsFont); 13: Chart1.View3d:=not Chart1.View3d:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Alignment:=laRight then  Chart1.Legend.Alignment:=laRigh	h;	or(False);
:=TeeDefHorizMargin+BevelWidth;  MarginRight :=TeeDefHorizMargin+BevelWidth;  MarginBottom:=TeeDefVerticalMargin+BevelWidth;  MarginBottom:=TeeDefVerticalMargin+BevelWidth;  cnd;  22: Chart1.Legend.Visible:=not Chart1.Legend.Visible;  23: if Random(10)<5 then Chart1.Legend.LegendStyle:=lsSeries clse Chart1.Legend.LegendStyle:=lsValues;  24: tmpSeries.Active:=not tmpSeries.Active:=not tmpSeries.Marks.BackColor:=Rand omColor(True);  26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible: 27: RandomPen(Chart1.BottomAxis.Grid);  28: RandomPen(Chart1.LeftAxis.Grid);  29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible;  30: RandomPen(Chart1.Legend.Frame);  16: Chart1.BottomAxis.Grid.Visible:=not Chart1.Legend.Alignment:=laRight then  Chart1.Legend.Alignment:=laRight;  11: RandomFont(Chart1.BottomAxis.La belsFont);  13: Chart1.View3d:=not Chart1.View3d:=not Chart1.View3d: 14: tmpSeries.BarWidthPercent:=50+R andom(40);  15: RandomFont(tmpSeries.Marks.Font );  16: Chart1.BottomAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible;  17: Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible;		•
MarginRight :=TeeDefHorizMargin+BevelWidth;  MarginBottom:=TeeDefVerticalMargin+BevelWidth;  end;  22: Chart1.Legend.Visible:=not Chart1.Legend.Visible;  23: if Random(10)<5 then Chart1.Legend.LegendStyle:=lsSeries else Chart1.Legend.LegendStyle:=lsValues;  24: tmpSeries.Active:=not tmpSeries.Active;  25: tmpSeries.Marks.BackColor:=Rand omColor(True);  26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible;  27: RandomPen(Chart1.BottomAxis.Grid);  28: RandomPen(Chart1.LeftAxis.Grid);  29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible;  30: RandomPen(Chart1.Legend.Frame);  10: if Chart1.Legend.Alignment:=laRight then  Chart1.Legend.Alignment:=laRight;  11: RandomFont(Chart1.BottomAxis.LabelsFont);  12: RandomFont(Chart1.LeftAxis.LabelsFont);  13: Chart1.View3d:=not Chart1.View3d: 14: tmpSeries.BarWidthPercent:=50+R andom(40);  15: RandomFont(tmpSeries.Marks.Font);  16: Chart1.BottomAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not	S .	Chart1.Chart3dPercent:=5+Rando
### Second Color of Chart 1. Legend Frame From Endom Font (Chart 1. Legend From Endom Font (Chart 1		
MarginBottom:=TeeDefVerticalMargin+BevelWidth; end; 22: Chart1.Legend.Visible:=not Chart1.Legend.Visible: 23: if Random(10)<5 then Chart1.Legend.LegendStyle:=IsSeries else Chart1.Legend.LegendStyle:=IsValues; 24: tmpSeries.Active:=not tmpSeries.Active; 25: tmpSeries.Marks.BackColor:=RandomColor(Chart1.LeftAxis.LabelsFont); 26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible: 27: RandomPen(Chart1.BottomAxis.Grid); 29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible: 30: RandomPen(Chart1.Legend.Frame); Chart1.BackColor:=RandomColor(False); 10: if Chart1.Legend.Alignment:=laRight then Chart1.Legend.Alignment:=laLeft else Chart1.Legend.Alignment:=laRight; 11: RandomFont(Chart1.BottomAxis.LabelsFont); 12: RandomFont(Chart1.LeftAxis.LabelsFont); 13: Chart1.View3d:=not Chart1.View3d: 14: tmpSeries.BarWidthPercent:=50+R andom(40); 15: RandomFont(tmpSeries.Marks.Font); 16: Chart1.BottomAxis.Grid.Visible:=not Chart1.BottomAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=n		· · ·
MarginBottom:=TeeDefVerticalMargin+BevelWidth; end;  22: Chart1.Legend.Visible:=not Chart1.Legend.Visible;  23: if Random(10)<5 then Chart1.Legend.LegendStyle:=lsSerie s else Chart1.Legend.LegendStyle:=lsValues;  24: tmpSeries.Active:=not tmpSeries.Active;  25: tmpSeries.Marks.BackColor:=RandomColor(True);  26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible;  27: RandomPen(Chart1.BottomAxis.Grid);  28: RandomPen(Chart1.LeftAxis.Grid);  29: Chart1.Legend.Alignment:=laRight then  Chart1.Legend.Alignment:=laRight the	· 1000011011011011111111111111111111111	
gin+BevelWidth; end;  22: Chart1.Legend.Visible:=not Chart1.Legend.Visible;  23: if Random(10)<5 then Chart1.Legend.LegendStyle:=lsSerie s else Chart1.Legend.LegendStyle:=lsValu es;  24: tmpSeries.Active:=not tmpSeries.Active;  25: tmpSeries.Marks.BackColor:=Rand omColor(True);  26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible;  27: RandomPen(Chart1.BottomAxis.Grid);  28: RandomPen(Chart1.LeftAxis.Grid);  29: Chart1.Legend.Alignment:=laRight then  Chart1.Legend.Alignment:=l	MarginRattom:-TooDafVarticalMar	•
end;  22: Chart1.Legend.Visible:=not Chart1.Legend.Visible;  23: if Random(10)<5 then Chart1.Legend.LegendStyle:=lsSerie s  else Chart1.Legend.LegendStyle:=lsValues;  24: tmpSeries.Active:=not tmpSeries.Active;  25: tmpSeries.Marks.BackColor:=Rand omColor(True);  26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible;  27: RandomPen(Chart1.BottomAxis.Grid);  28: RandomPen(Chart1.LeftAxis.Grid);  29: Chart1.Legend.Alignment:=laRight then  Chart1.Legend.Alignment:=laRight else  Chart1.Legend.Alignment:=laRight then  Chart1.Legend.Alignment:=laLeft else  Chart1.Legend.Alignment:=laLeft else  Chart1.Legend.Alignment:=laLeft else  Chart1.Legend.Alignment:=laRight;  11: RandomFont(Chart1.BottomAxis.LabelsFont);  12: RandomFont(Chart1.LeftAxis.LabelsFont);  13: Chart1.View3d:=not Chart1.View3d;  14: tmpSeries.BarWidthPercent:=50+R andom(40);  15: RandomFont(tmpSeries.Marks.Font);  16: Chart1.BottomAxis.Grid.Visible:=not Chart1.BottomAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.Legend.Alignment:=laRight then  Chart1.Legend.Alignment:=laCeft else  11: RandomFont(Chart1.BottomAxis.LabelsFont);  12: RandomFont(Chart1.BottomAxis.LabelsFont);  13: Chart1.View3d;  14: tmpSeries.BarWidthPercent:=50+R andom(40);  15: RandomFont(tmpSeries.Marks.Font);  16: Chart1.BottomAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not C		
22: Chart1.Legend.Visible:=not Chart1.Legend.Visible;  23: if Random(10)<5 then Chart1.Legend.LegendStyle:=lsSerie s  else Chart1.Legend.LegendStyle:=lsValu es;  24: tmpSeries.Active:=not tmpSeries.Marks.BackColor:=Rand omColor(True);  26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible;  27: RandomPen(Chart1.BottomAxis.Gri d);  28: RandomPen(Chart1.LeftAxis.Grid);  29: Chart1.Legend.Alignment:=laRight;  11: RandomFont(Chart1.BottomAxis.La belsFont);  12: RandomFont(Chart1.LeftAxis.Label sFont);  13: Chart1.View3d:=not Chart1.View3d;  14: tmpSeries.BarWidthPercent:=50+R andom(40);  15: RandomFont(tmpSeries.Marks.Font);  15: Chart1.Legend.Alignment:=laRight;  11: RandomFont(Chart1.BottomAxis.La belsFont);  12: RandomFont(Chart1.LeftAxis.Label sFont);  13: Chart1.View3d:=not Chart1.View3d;  14: tmpSeries.BarWidthPercent:=50+R andom(40);  15: RandomFont(tmpSeries.Marks.Font);  16: Chart1.BottomAxis.Grid.Visible:=not Chart1.BottomAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not		
Chart1.Legend.Visible;  23: if Random(10) < 5 then Chart1.Legend.LegendStyle:=lsSerie s  else Chart1.Legend.LegendStyle:=lsValu es;  24: tmpSeries.Active:=not tmpSeries.Marks.BackColor:=Rand omColor(True);  25: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible;  27: RandomPen(Chart1.BottomAxis.Grid);  28: RandomPen(Chart1.LeftAxis.Grid);  29: Chart1.Legend.Alignment:=laRight;  11: RandomFont(Chart1.BottomAxis.LabelsFont);  12: RandomFont(Chart1.BottomAxis.LabelsFont);  13: Chart1.View3d:=not Chart1.View3d:=not Chart1.View3d:  14: tmpSeries.BarWidthPercent:=50+R andom(40);  15: RandomFont(tmpSeries.Marks.Font);  16: Chart1.BottomAxis.Grid.Visible:=not Chart1.Legend.Frame.Visible;  17: Chart1.LeftAxis.Grid.Visible:=not	,	0 0
23: if Random(10)<5 then Chart1.Legend.LegendStyle:=lsSerie s  else Chart1.Legend.LegendStyle:=lsValu es;  24: tmpSeries.Active:=not tmpSeries.Marks.BackColor:=Rand omColor(True);  26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible;  27: RandomPen(Chart1.BottomAxis.Grid);  28: RandomPen(Chart1.LeftAxis.Grid);  29: Chart1.Legend.Frame.Visible:=not tChart1.Legend.Frame.Visible;  30: RandomPen(Chart1.Legend.Frame);  Chart1.Legend.Alignment:=laLeft else  Chart1.Legend.Alignment:=laLeft else  Chart1.Legend.Alignment:=laLeft else  Chart1.Legend.Alignment:=laLeft else  Chart1.Legend.Alignment:=laLeft else  Chart1.Legend.Alignment:=laCeft else  Chart1.Legend.Alignment:=laLeft else  Chart1.Legend.Alignment:=laCeft else  Chart1.Legend.Alignment:=laCeft else  Chart1.Legend.Alignment:=laCeft else  Chart1.Legend.Alignment:=laCeft else  Chart1.Legend.Alignment:=laCeft else  Chart1.Legend.Alignment:=laCeft else  11: RandomFont(Chart1.BottomAxis.LabelsFont); 13: Chart1.View3d:=not Chart1.View3d; 14: tmpSeries.BarWidthPercent:=50+R andom(40); 15: RandomFont(tmpSeries.Marks.Font); 16: Chart1.BottomAxis.Grid.Visible:=not t Chart1.BottomAxis.Grid.Visible:=not t Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=no	<u> </u>	then
Chart1.Legend.LegendStyle:=lsSerie s  else Chart1.Legend.LegendStyle:=lsValu es;  24: tmpSeries.Active:=not tmpSeries.Marks.BackColor:=Rand omColor(True);  26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible;  27: RandomPen(Chart1.BottomAxis.Grid);  28: RandomPen(Chart1.LeftAxis.Grid);  29: Chart1.Legend.Alignment:=laRight;  11: RandomFont(Chart1.BottomAxis.LabelsFont);  12: RandomFont(Chart1.LeftAxis.LabelsFont);  13: Chart1.View3d:=not Chart1.View3d:=not Chart1.View3d;  14: tmpSeries.BarWidthPercent:=50+R andom(40);  15: RandomFont(tmpSeries.Marks.Font);  16: Chart1.BottomAxis.Grid.Visible:=not Chart1.BottomAxis.Grid.Visible:=not Chart1.BottomAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not		
else Chart1.Legend.LegendStyle:=lsValu es;  24: tmpSeries.Active:=not tmpSeries.Marks.BackColor:=Rand omColor(True);  26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible;  27: RandomPen(Chart1.BottomAxis.Gri d);  28: RandomPen(Chart1.LeftAxis.Grid);  29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible;  30: RandomPen(Chart1.Legend.Frame);  Chart1.Legend.Alignment:=laRight;  11: RandomFont(Chart1.BottomAxis.La belsFont);  12: RandomFont(Chart1.LeftAxis.Label sFont);  13: Chart1.View3d:=not Chart1.View3d;  14: tmpSeries.BarWidthPercent:=50+R andom(40);  15: RandomFont(tmpSeries.Marks.Font) );  16: Chart1.BottomAxis.Grid.Visible:=not t Chart1.BottomAxis.Grid.Visible:=not t Chart1.LeftAxis.Grid.Visible:=not	` ,	
chart1.Legend.LegendStyle:=lsValues;  24: tmpSeries.Active:=not tmpSeries.Marks.BackColor:=RandomColor(True);  26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible;  27: RandomPen(Chart1.BottomAxis.Grid);  28: RandomPen(Chart1.LeftAxis.Grid);  29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible;  30: RandomPen(Chart1.Legend.Frame);  Chart1.Legend.Alignment:=laRight;  11: RandomFont(Chart1.BottomAxis.LabelsFont);  12: RandomFont(Chart1.LeftAxis.LabelsFont);  13: Chart1.View3d:=not Chart1.View3d;  14: tmpSeries.BarWidthPercent:=50+R andom(40);  15: RandomFont(tmpSeries.Marks.Font);  16: Chart1.BottomAxis.Grid.Visible:=not tChart1.BottomAxis.Grid.Visible:=not tChart1.BottomAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not C	Chart1.Legend.LegendStyle:=lsSerie	else
Chart1.Legend.LegendStyle:=IsValu es;  24: tmpSeries.Active:=not tmpSeries.Marks.BackColor:=Rand omColor(True);  26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible;  27: RandomPen(Chart1.BottomAxis.Gri d);  28: RandomPen(Chart1.LeftAxis.Grid);  29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible;  30: RandomPen(Chart1.Legend.Frame) ;  Chart1.Legend.Frame  Chart1.LeftAxis.Grid.Visible:=not		
es; 24: tmpSeries.Active:=not tmpSeries.Active; 25: tmpSeries.Marks.BackColor:=Rand omColor(True); 26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible; 27: RandomPen(Chart1.BottomAxis.Grid); 28: RandomPen(Chart1.LeftAxis.Grid); 29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible; 30: RandomPen(Chart1.Legend.Frame); RandomPen(Chart1.Legend.Frame);  RandomPen(Chart1.Legend.Frame); Chart1.Legend.Frame.Visible; 12: RandomFont(Chart1.LeftAxis.Label sFont); 13: Chart1.View3d:=not Chart1.Vie		
24: tmpSeries.Active:=not tmpSeries.Active;  25: tmpSeries.Marks.BackColor:=Rand omColor(True);  26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible;  27: RandomPen(Chart1.BottomAxis.Gri d);  28: RandomPen(Chart1.LeftAxis.Grid);  29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible;  30: RandomPen(Chart1.Legend.Frame) ;  belsFont);  12: RandomFont(Chart1.LeftAxis.Label sFont);  13: Chart1.View3d:=not Chart1.View3d;  14: tmpSeries.BarWidthPercent:=50+R andom(40);  15: RandomFont(tmpSeries.Marks.Font );  16: Chart1.BottomAxis.Grid.Visible:=not t Chart1.BottomAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible;  17: Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible;	Chart1.Legend.LegendStyle:=lsValu	11:
tmpSeries.Active;  25: tmpSeries.Marks.BackColor:=Rand omColor(True);  26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible;  27: RandomPen(Chart1.BottomAxis.Gri d);  28: RandomPen(Chart1.LeftAxis.Grid);  29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible;  30: RandomPen(Chart1.Legend.Frame);  Chart1.Legend.Frame.Visible;  30: Chart1.Legend.Frame.Visible;  12: RandomFont(Chart1.LeftAxis.Label sFont);  13: Chart1.View3d:=not Chart1.View3d;  14: tmpSeries.BarWidthPercent:=50+R andom(40);  15: RandomFont(tmpSeries.Marks.Font );  16: Chart1.BottomAxis.Grid.Visible:=not t Chart1.BottomAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Vis	es;	RandomFont(Chart1.BottomAxis.La
25: tmpSeries.Marks.BackColor:=Rand omColor(True); 26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible; 27: RandomPen(Chart1.BottomAxis.Gri d); 28: RandomPen(Chart1.LeftAxis.Grid); 29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible; 30: RandomPen(Chart1.Legend.Frame);  RandomFont(Chart1.LeftAxis.Label sFont); 13: Chart1.View3d:=not Chart1.View3d; 14: tmpSeries.BarWidthPercent:=50+R andom(40); 15: RandomFont(tmpSeries.Marks.Font ); 16: Chart1.BottomAxis.Grid.Visible:=not t Chart1.BottomAxis.Grid.Visible:=not t Chart1.LeftAxis.Grid.Visible:=not	24: tmpSeries.Active:=not	belsFont);
tmpSeries.Marks.BackColor:=Rand omColor(True);  26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible;  27: RandomPen(Chart1.BottomAxis.Gri d);  28: RandomPen(Chart1.LeftAxis.Grid);  29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible;  30: RandomPen(Chart1.Legend.Frame);  Chart1.Legend.Frame.Visible;  30: Chart1.Legend.Frame);  Chart1.Legend.Frame.Visible;  13: Chart1.View3d:=not Chart1.View3	tmpSeries.Active;	12:
omColor(True); 26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible; 27: RandomPen(Chart1.BottomAxis.Gri d); 28: RandomPen(Chart1.LeftAxis.Grid); 29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible; 30: RandomPen(Chart1.Legend.Frame);  13: Chart1.View3d:=not Chart1.View3d; 14: tmpSeries.BarWidthPercent:=50+R andom(40); 15: RandomFont(tmpSeries.Marks.Font ); 16: Chart1.BottomAxis.Grid.Visible:=not t Chart1.BottomAxis.Grid.Visible:=not t Chart1.LeftAxis.Grid.Visible:=not	25:	RandomFont(Chart1.LeftAxis.Label
omColor(True);13: Chart1.View3d:=not26: tmpSeries.Marks.Visible:=notChart1.View3d;14:14:27:tmpSeries.BarWidthPercent:=50+RRandomPen(Chart1.BottomAxis.Grid);andom(40);28:RandomFont(tmpSeries.Marks.Font);29:RandomFont(tmpSeries.Marks.Font);29:Chart1.Legend.Frame.Visible:=notChart1.Legend.Frame.Visible;Chart1.BottomAxis.Grid.Visible:=not30:17:RandomPen(Chart1.Legend.Frame);Chart1.LeftAxis.Grid.Visible:=notChart1.LeftAxis.Grid.Visible;	tmpSeries.Marks.BackColor:=Rand	sFont);
26: tmpSeries.Marks.Visible:=not tmpSeries.Marks.Visible;  27: RandomPen(Chart1.BottomAxis.Gri d);  28: RandomPen(Chart1.LeftAxis.Grid);  29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible;  30: RandomPen(Chart1.Legend.Frame);  Chart1.View3d;  14: tmpSeries.BarWidthPercent:=50+R andom(40);  15: RandomFont(tmpSeries.Marks.Font);  Chart1.BottomAxis.Grid.Visible:=not t Chart1.BottomAxis.Grid.Visible:=not t Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible;	_	13: Chart1.View3d:=not
tmpSeries.Marks.Visible;  27: RandomPen(Chart1.BottomAxis.Grid);  28: RandomPen(Chart1.LeftAxis.Grid);  29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible;  30: RandomPen(Chart1.Legend.Frame);  Chart1.Legend.Frame.Visible;  Chart1.Legend.Frame.Visible;  Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible;	` '/'	
27: RandomPen(Chart1.BottomAxis.Grid); 28: RandomPen(Chart1.LeftAxis.Grid); 29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible; 30: RandomPen(Chart1.Legend.Frame);  Chart1.Legend.Frame);  Chart1.Legend.Frame);	<u>-</u>	
RandomPen(Chart1.BottomAxis.Grid);  28: RandomPen(Chart1.LeftAxis.Grid);  29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible;  30: RandomPen(Chart1.Legend.Frame) ;  Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible;		
d); 28: RandomPen(Chart1.LeftAxis.Grid); 29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible; 30: RandomPen(Chart1.Legend.Frame) ; Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible;		-
28: RandomPen(Chart1.LeftAxis.Grid); 29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible; 30: RandomPen(Chart1.Legend.Frame) ;  Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:	`	
RandomPen(Chart1.LeftAxis.Grid); 29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible; 30: RandomPen(Chart1.Legend.Frame) ;  Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible;		
29: Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible; 30: RandomPen(Chart1.Legend.Frame) ;  16: Chart1.BottomAxis.Grid.Visible:=not t Chart1.BottomAxis.Grid.Visible; 17: Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible;		` -
Chart1.Legend.Frame.Visible:=not Chart1.Legend.Frame.Visible;  30: RandomPen(Chart1.Legend.Frame) ;  Chart1.BottomAxis.Grid.Visible:=not t Chart1.BottomAxis.Grid.Visible:=not t Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible:=not		7.
Chart1.Legend.Frame.Visible; 30: RandomPen(Chart1.Legend.Frame) ;  t Chart1.BottomAxis.Grid.Visible; 17: Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible;		
30: RandomPen(Chart1.Legend.Frame) ;  17: Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible;	S	
RandomPen(Chart1.Legend.Frame) ; Chart1.LeftAxis.Grid.Visible:=not Chart1.LeftAxis.Grid.Visible;		·
; Chart1.LeftAxis.Grid.Visible;		
	RandomPen(Chart1.Legend.Frame)	
31:	;	-
101	31:	18:

49:	RandomPen(tmpSeries.BarPen);
RandomBrush(tmpSeries.BarBrush)	32:
;	tmpSeries.ColorEachPoint:=not
50:	tmpSeries.ColorEachPoint;
Chart1.Title.Frame.Visible:=not	33:
Chart1.Title.Frame.Visible;	Chart1.Title.Alignment:=TAlignmen
51:	t(Random(3));
RandomPen(Chart1.Title.Frame);	34:
52:	RandomFont(Chart1.Title.Font);
Chart1.Foot.Frame.Visible:=not	35:
Chart1.Foot.Frame.Visible;	RandomFont(Chart1.Legend.Font);
53:	36:
RandomPen(Chart1.Foot.Frame);	Chart1.Legend.Color:=RandomColo
54:	r(False);
Chart1.Title.AdjustFrame:=not	37:
Chart1.Title.AdjustFrame;	
•	Chart1.Legend.TextStyle:=TLegend
55: Chart1 Fact AdjustEromormat	TextStyle(Random(5));
Chart1.Foot.AdjustFrame:=not	38:
Chart1.Foot.AdjustFrame;	Chart1.Legend.TopPos:=5+Random
56:	(90);
RandomBrush(Chart1.Title.Brush);	39:
57:	RandomPen(Chart1.BottomAxis.Axi
RandomBrush(Chart1.Foot.Brush);	s);
58:	40:
Chart1.BottomAxis.MinorTickLengt	RandomPen(Chart1.LeftAxis.Axis);
h:=Random(8);	41:
59:	RandomPen(Chart1.BottomAxis.Tic
Chart1.LeftAxis.MinorTickLength:	ks);
=Random(8);	42:
60:	RandomPen(Chart1.LeftAxis.Ticks)
RandomPen(Chart1.BottomAxis.Mi	:
norTicks);	43:
61:	RandomPen(Chart1.BottomAxis.Tic
RandomPen(Chart1.LeftAxis.Minor	ksInner);
Ticks);	44:
62:	RandomPen(Chart1.LeftAxis.TicksI
RandomPen(Chart1.LeftWall.Pen);	nner);
63:	45:
RandomPen(Chart1.BottomWall.Pe	Chart1.BottomAxis.TickLength:=Ra
n);	ndom(8);
64:	46:
Chart1.LeftWall.Color:=RandomCo	Chart1.LeftAxis.TickLength:=Rand
lor(False);	om(8);
65:	47:
Chart1.BottomWall.Color:=Random	Chart1.BottomAxis.TickInnerLengt
Color(False);	h:=Random(8);
66:	48:
Chart1.LeftAxis.MinorTickCount:=	Chart1.LeftAxis.TickInnerLength:=
1+Random(6);	Random(8);

	67:
39	Chart1.BottomAxis.MinorTickCoun
{**********	t:=1+Random(6);
********	68:
{ TeeChart Delphi Component	Chart1.LeftAxis.Title.Angle:=Rando
Library }	mAngle;
{ Table Data-Aware Chart Demo	69:
}	Chart1.BottomAxis.Title.Angle:=Ra
{ Copyright (c) 1995-2001 by David	ndomAngle;
Berneda }	70:
{ All rights reserved }	Chart1.LeftAxis.LabelsAngle:=Rand
{*************************************	omAngle;
*********	71:
unit tablepie;	Chart1.BottomAxis.LabelsAngle:=R
•	andomAngle;
interface	end;
	{ re-start timer }
uses	Timer1.Enabled:=True;
SysUtils, WinTypes, WinProcs,	end;
Messages, Classes, Graphics,	
Controls,	<> procedure
Forms, Dialogs, StdCtrls, Buttons,	TStackedForm.Shape1MouseUp(Sen
ExtCtrls, Grids, DBGrids, DB,	der: TObject; Button:
DBTables, Chart, Series, DbChart,	TMouseButton;
Teengine, TeeProcs;	Shift: TShiftState; X, Y: Integer);
	begin
type	{ Run the Color dialog to change
TTablePieForm = class(TForm)	Series color }
DBChart1: TDBChart;	With
PieSeries1: TPieSeries;	Chart1.Series[ComboBox1.ItemInde
Table1: TTable;	x] do
DataSource1: TDataSource;	
DBGrid1: TDBGrid;	SeriesColor:=EditColor(Self,SeriesC
Panel1: TPanel;	olor);
Table1NAME: TStringField;	RefreshShape;
Table1SIZE: TSmallintField;	end;
Table1WEIGHT: TSmallintField;	
CheckBox1: TCheckBox;	<> procedure
RadioGroup1: TRadioGroup;	TStackedForm.CheckBox2Click(Sen
BitBtn1: TBitBtn;	der: TObject);
<> procedure	begin
CheckBox1Click(Sender: TObject);	{ Turn on / off Chart 3D }
<> procedure	
RadioGroup1Click(Sender:	Chart1.View3D:=CheckBox2.Check
TObject);	ed;
<> procedure	end;
PieSeries1Click(Sender:	
TChartSeries; ValueIndex: Integer;	end.
Button: TMouseButton; Shift:	

Y: Integer);	TShiftState; X, Y: Integer);
Const CarriageReturn=#13#10;	private
begin	{ Private declarations }
{ On Clicked Pie, show the user the	public
pie data }	{ Public declarations }
With PieSeries1 do	end;
ShowMessage( XLabel[ValueIndex]	enu,
+ CarriageReturn+	implementation
Carriageneturn	Implementation
MarkPercent(ValueIndex,True));	{\$R *.dfm}
end;	tak .um;
- Circly	<> procedure
end.	TTablePieForm.CheckBox1Click(Se
	nder: TObject);
40	begin
{***********	negin
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	Table1.Active:=CheckBox1.Checked
{ TeeChart Delphi Component	; { < open / close the table }
Library 3.0 }	end;
{ Demo }	ciu,
{ Copyright (c) 1995-2001 by David	<> procedure
Berneda }	TTablePieForm.RadioGroup1Click(
{ All rights reserved }	Sender: TObject);
{***********	begin
\ **********	{ change the Pie value source (the
unit teemain;	table field name) }
	Case RadioGroup1.ItemIndex of
interface	0:
	PieSeries1.PieValues.ValueSource:='
uses	Size';
SysUtils, WinTypes, WinProcs,	1:
Messages, Classes, Graphics,	PieSeries1.PieValues.ValueSource:='
Controls,	Weight';
Forms, Dialogs, ExtCtrls,	end;
TeEngine, Series, TeeProcs, Chart,	{ change the Chart Foot text to
StdCtrls,	display the current table Field }
TeeFunci;	*
,	DBChart1.Foot.Text[1]:='PieValueS
type	ource:=Table1'+
TTeeMainForm = class(TForm)	
Panel1: TPanel;	PieSeries1.PieValues.ValueSource;
Button1: TButton;	end;
Image1: TImage;	
Timer1: TTimer;	<> procedure
Button2: TButton;	TTablePieForm.PieSeries1Click(Sen
Button3: TButton;	der: TChartSeries;
Button4: TButton;	ValueIndex: Integer; Button:
Button6: TButton;	TMouseButton; Shift: TShiftState;
Chart2: TChart;	Χ,
Churta, I Churt,	

Putton/Click(Sandon TObicat).	Fast InaCarias 1. TEast I inaCarias
Button4Click(Sender: TObject);	FastLineSeries1: TFastLineSeries;
<> procedure	FastLineSeries2: TFastLineSeries;
Button6Click(Sender: TObject);	Timer2: TTimer;
<> procedure	Label11: TLabel;
Label3Click(Sender: TObject);	Label12: TLabel;
private	Panel2: TPanel;
{ Private declarations }	Chart1: TChart;
DeltaZoom:Integer;	Series1: TPieSeries;
public	Chart7: TChart;
{ Public declarations }	PieSeries6: TPieSeries;
<> procedure	Chart8: TChart;
ShowForm(FormClass:	PieSeries7: TPieSeries;
TFormClass);	Chart9: TChart;
end;	PieSeries8: TPieSeries;
	Chart10: TChart;
var	PieSeries9: TPieSeries;
TeeMainForm: TTeeMainForm;	Label13: TLabel;
	Label15: TLabel;
implementation	Label16: TLabel;
	Label2: TLabel;
{\$R *.dfm}	Panel3: TPanel;
	Label1: TLabel;
Uses TeeAbout, TeeBasic, Features,	Label9: TLabel;
UDemoCha, Specs, ShellAPI;	Label3: TLabel;
	<> procedure
<> procedure	FormCreate(Sender: TObject);
TTeeMainForm.ShowForm(FormCl	<> procedure
ass: TFormClass);	Timer1Timer(Sender: TObject);
begin	<> procedure
Timer1.Enabled:=False;	Button1Click(Sender: TObject);
Timer2.Enabled:=False;	<> procedure
With FormClass.Create(Self) do	Timer2Timer(Sender: TObject);
try	<> procedure
ShowModal;	Button2Click(Sender: TObject);
finally	<> procedure
Free;	Chart2MouseUp(Sender: TObject;
end;	Button: TMouseButton;
Timer2.Enabled:=True;	,
Timer1.Enabled:=True;	Shift: TShiftState; X, Y:
	Integer);
end;	<> procedure Chart2MaysaDayn(Sandar)
	Chart2MouseDown(Sender:
<> procedure	TObject; Button: TMouseButton;
TTeeMainForm.FormCreate(Sender	Shift: TShiftState; X, Y:
: TObject);	Integer);
begin	<> procedure
TeeEraseBack:=False;	Label10Click(Sender: TObject);
PieSeries6.FillSampleValues(5);	<> procedure
PieSeries9.FillSampleValues(5);	Button3Click(Sender: TObject);
PieSeries7.FillSampleValues(5);	<> procedure

TTeeMainForm.Timer2Timer(Send	Series1.FillSampleValues(5);
er: TObject);	PieSeries8.CheckDataSource;
var tmpX:Double;	1 reserveso. eneckbatasource,
begin	FastLineSeries1.FillSampleValues(2
With FastLineSeries1 do	0);
begin	0),
tmpX:=XValues[1]-XValues[0];	FastLineSeries2.FillSampleValues(2
Delete(0);	0);
Detect(0);	DeltaZoom:=5;
AddXY(XValues.Last+tmpX,Rando	end;
m(100),",clTeeColor);	Citu,
end;	<> procedure
With FastLineSeries2 do	TTeeMainForm.Timer1Timer(Send
begin	er: TObject);
tmpX:=XValues[1]-XValues[0];	var tmp:Integer;
Delete(0);	begin
Detection,	PieSeries6.Rotate(5);
AddXY(XValues.Last+tmpX,Rando	PieSeries7.Rotate(5);
m(100),",clTeeColor);	PieSeries9.Rotate(5);
end;	Series1.Rotate(355);
end;	PieSeries8.Rotate(355);
enu,	With Chart9.View3DOptions do
/ > nwooodure	begin
<> procedure TTeeMainForm.Button2Click(Sende	Zoom:=Zoom-DeltaZoom;
r: TObject);	if (Zoom<60) or (Zoom>110) then
begin	DeltaZoom:=-DeltaZoom;
ShowForm(TDemoForm);	end;
end;	With PieSeries8 do
enu,	begin
<> procedure	tmp:=Random(Count);
TTeeMainForm.Chart2MouseUp(Se	if ExplodedSlice[tmp]>20 then
nder: TObject; Button:	ii ExplodedSitce[tinp]>20 then
TMouseButton;	ExplodedSlice[tmp]:=ExplodedSlice[
Shift: TShiftState; X, Y: Integer);	tmp]-1
begin	else
Label10Click(Self);	CISC
end;	ExplodedSlice[tmp]:=ExplodedSlice[
ciru,	tmp]+1;
<> procedure	end;
TTeeMainForm.Chart2MouseDown	end;
(Sender: TObject;	cnu,
Button: TMouseButton; Shift:	<> procedure
TShiftState; X, Y: Integer);	TTeeMainForm.Button1Click(Sende
begin	r: TObject);
Chart2.BevelOuter:=bvLowered;	begin
end;	Close;
,	end;
<> procedure	Í
TTeeMainForm.Label10Click(Sende	<> procedure
	p. 000 man 0

r: TObject);
begin
ShowForm(TTeeAboutForm);
Chart2.BevelOuter:=bvRaised;
end;
<> procedure
TTeeMainForm.Button3Click(Sende
r: TObject);
begin
ShowForm(TFeaturesForm);
end;
<> procedure
TTeeMainForm.Button4Click(Sende
r: TObject);
begin
ShowForm(TChartSpecs);
end;
<> procedure
TTeeMainForm.Button6Click(Sende
r: TObject);
begin
ShowForm(TSeriesForm);
end;
<> procedure
TTeeMainForm.Label3Click(Sender
: TObject);
Var St:Array[0255] of char;
begin
Timer1.Enabled:=False;
Timer2.Enabled:=False;
ShellExecute(0,'open',StrPCopy(St,'
http://'+Label3.Caption+
'/products/teechart/delphi6'),nil,nil,S
W_SHOW);
end;
end.
\\\\\\\\
The end
ALAA EDDIN LUBBAD
00963944575371